

ESD-TR-68-134

MTR-564

ESD RECORD COPY

RETURN TO
SCIENTIFIC & TECHNICAL INFORMATION DIVISION
(ESTI), BUILDING 1211

ESD ACCESSION LIST

ESTI Call No. 63870

Copy No. 1 of 2 cys.

PROGRAM/CODING SPECIFICATION FOR THE AFICCS REPORT GENERATION (REPGEN) CAPABILITY

JANUARY 1969

N. B. Sutherland

Prepared for

DIRECTOR OF PLANNING AND TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts



This document has been approved for public release and sale; its distribution is unlimited.

Project 512V
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract AF19(628)-5165

A0681840

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Do not return this copy. Retain or destroy.

PROGRAM/CODING SPECIFICATION FOR THE AFICCS REPORT
GENERATION (REPGEN) CAPABILITY

JANUARY 1969

N. B. Sutherland

Prepared for

DIRECTOR OF PLANNING AND TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts



This document has been approved for public release and sale; its distribution is unlimited.

Project 512V
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract AF19(625)-5165

FOREWORD

The work reported in this document was performed by The MITRE Corporation, Bedford, Massachusetts, for the Director of Planning and Technology, Electronic Systems Division, of the Air Force Systems Command under Contract AF 19(628)-5165.

REVIEW AND APPROVAL

This technical report has been reviewed and is approved.

WILLIAM F. HEISLER, COL, USAF
Chief, Command Systems Division
Directorate of Planning & Technology

ABSTRACT

The Report Generation (REPGEN) capability comprises a set of programs designed to run on an IBM 1410 computer under the Air Force Integrated Command and Control System (AFICCS). The primary purpose of REPGEN is to provide a capability for producing a printed report from a file of input data contained on punched cards, magnetic tape or on the system disk. This document contains detailed information regarding the structure and operation of the REPGEN programs, including table descriptions, core maps, program flow-charts and abstracts.

TABLE OF CONTENTS

	<u>Page</u>
LIST OF ILLUSTRATIONS	vii
LIST OF TABLES	vii
INTRODUCTION	1
OVERVIEW	2
Table Generation	2
LID (Load Input Descriptors)	2
SAVER	2
LEGVAL (Legitimate Value Loader)	4
LOD (Load Output Descriptors)	4
File Generation	4
Report Production	5
TABLES	6
Report Descriptor Table	6
Record Descriptor Table	7
Attribute Descriptor Table	8
Selected Output Table (SOT)	9
Output Entry Descriptor Table (OEDT)	10
Legitimate Value Attribute Name Table	13
Legitimate Value Table	14
Condensed Output Program	16
Saved Statement Table Directory	18
Saved Statement Table	19
I/O Communication Record	21
OUTPUT PHASE COMMUNICATIONS ZONE	22
SYMBOLIC DISK ADDRESSES	26
REPGEN FLOWCHARTS AND ABSTRACTS	28
LID	28
LEGVAL	43
SAVER	50
LOD	53
REPGEN	89
RGPBYP	103
RGPOUT	107
LNINT	110

TABLE OF CONTENTS (CONCLUDED)

	<u>Page</u>
UPSTAT	113
PRINTO	116
CARDO	118
TAPEO	121
REALO	125
REMARK	146
REPEAT	149
REKALL	151
DELEET	153
PRTCRD	155
PRTTAP	157
RPTCRD	159

LIST OF ILLUSTRATIONS

<u>Figure Number</u>		<u>Page</u>
1	Overview of the Report Generation Process	3
2	Tables Associated With One Report	11
	Descriptor Table Entry	
3	LID Input and Output, Mode 1	12
4	LEGVAL Input and Output	15
5	LOD Input and Output	17
6	SAVER Input and Output	20
7	REPGEN Output Phase Core Map	25

SECTION I

INTRODUCTION

The purpose of this document is to present detailed information regarding the structure and operation of the various programs which comprise the REPGEN capability. It is assumed that the reader is familiar with the contents of the REPGEN User's Guide.

SECTION II

OVERVIEW

The Report Generation (REPGEN) capability comprises a set of programs designed to run on an IBM 1410 computer under the standard AFICCS system. The primary purpose of REPGEN is to provide a capability for producing a printed report from a file of input data contained on punched cards, on magnetic tape or on the system disk. The output may be optionally on magnetic tape or cards. The output format is controlled by use of a quasi-English language of limited vocabulary.

There are three major activities required in order to move from source data to finished report. These are:

- a. Table generation
- b. File generation
- c. Report production

Figure 1 shows the basic elements involved in each of these activities.

TABLE GENERATION

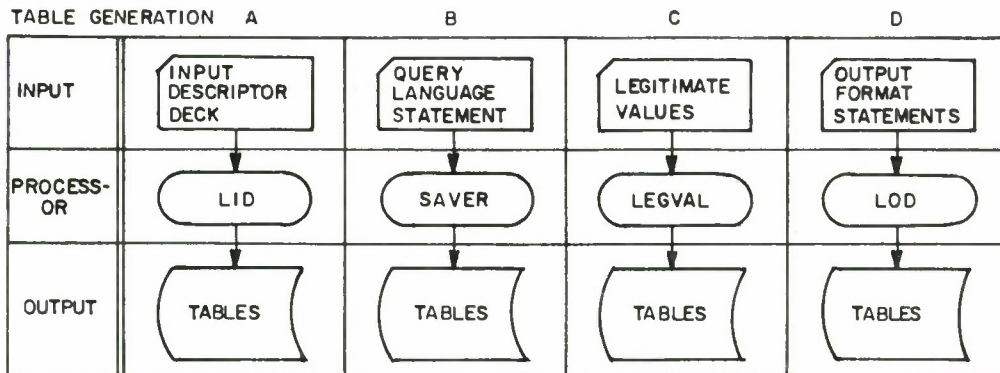
Before a report can be generated, certain basic information must be loaded into tables and placed on the system disk, to be available for File generation and Report production. The total REPGEN capability includes a number of support programs which are used to perform the necessary table construction.

LID (Load Input Descriptors)

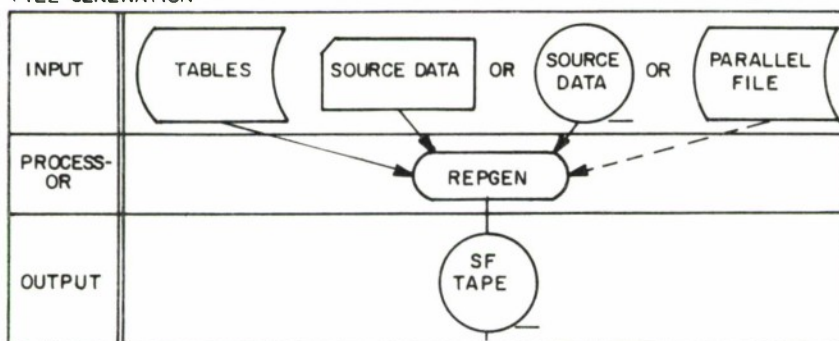
The input to LID is primarily a description of the source data which will be used to make up the serial file tape (SF tape) which in turn will be the input to the output phase of REPGEN. The various source data record types must be identified, and the data attributes associated with each record type must be defined in terms of length, position within the record, data type, etc.

SAVER

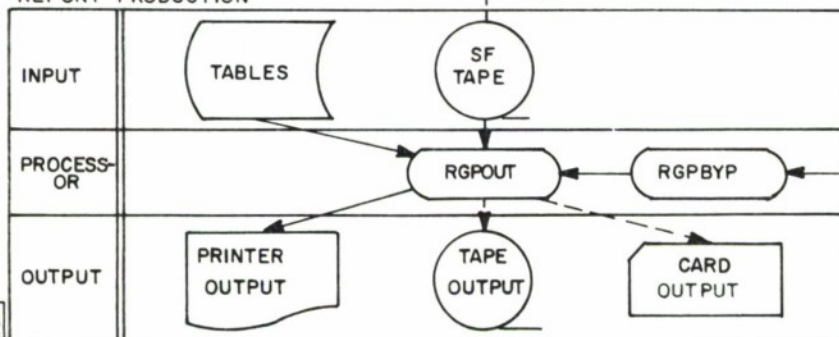
In some cases, the source data may reside on the AFICCS system disk in parallel file form. In order to produce a serial file tape (SF Tape) which can be used by REPGEN output programs, a Query Language retrieval must be performed against the parallel file. The Query Language retrieval statement must specify GETTAC as the process director. GETTAC (a variant of GETB) outputs the retrieved data on an SF Tape.



FILE GENERATION



REPORT PRODUCTION



CAN BE
RUN AS
ONE JOB

OR

CAN BE
ENTERED HERE
IF TAPE HAS
ALREADY BEEN
GENERATED

IB-26,237

Figure 1. Overview of the Report Generation Process

SAVER's primary function is to save a Query Language retrieval statement so that it may be called up for execution when required. (SAVER is a variant of SAVE.)

LEGVAL (Legitimate Value Loader)

As information is transferred from source data to the serial file tape, it may be desirable to check certain attributes for legality. This best applies to attributes which can legally take on one of a limited number of values.

LEGVAL produces a table which contains all of the allowable values for each attribute which is to be legality checked.

LOD (Load Output Descriptors)

A quasi-English language of limited vocabulary has been implemented to give the user the capability to structure a report. Using the language provided, data can be selected from the serial file tape, arithmetically manipulated, and arranged for presentation on a printed page. (Output may optionally be to magnetic tape or punched cards.) The set of statements which direct this activity may be called an Output Program.

An output program is initially processed by LOD, which performs syntax checking, and then eliminates all redundant characters in each statement. LOD also replaces attribute names with their length and position on the serial file tape. The resulting condensed output program is placed in a table on the system disk, to be available for later execution.

FILE GENERATION

The purpose of the file generation activity is to produce a serial file tape containing the data from which the desired report(s) will be generated. The source data from which the serial file tape is made may be in punched cards, on magnetic tape, or may reside on the system disk in parallel file form.

When the source data is in cards or on tape, we are said to be operating in Mode 1; when the source data is a parallel file on disk, we are operating in Mode 2. There are internal differences in the operation of the file generation program (REPGEN), depending upon the mode in which it runs. The end result, however, is essentially the same; a serial file tape is produced which contains the data to be used in generating the desired report(s).

REPORT PRODUCTION

Once the required tables have been generated and a serial file tape has been prepared, we are ready to produce a report. This is accomplished by operating what is called the Output Phase.

The output phase can proceed as a continuation of a file generation run; at the completion of file generation the computer operator is given the option to terminate or to continue with the output phase.

A serial file tape which was produced by a previous run can be used to generate a report by operating the output phase via the Bypass Mode. In this mode, file generation is bypassed and the output phase is immediately started.

During the output phase, the condensed output program is read from disk and the statements are interpreted and executed to produce the desired output. Up to nine additional copies of a printed report may be produced automatically.

SECTION III

TABLES

REPORT DESCRIPTOR TABLE

Created By: LID

Purpose: Each entry in this table identifies by name and number an active REPGEN report.

Table Size: 1 record (expandable) of 2800 characters.

Entry Size: 100 characters per entry, recorded in Move mode.

End of Table: A record mark in the first position of an entry indicates the end of the table.

Entry Description:

<u>Positions</u>	<u>Contents</u>
1-3	Report Number
4-15	Report Title
16	Input Media
17	Classification
18-47	Header Label
48-49	Size of Header Label
50-52	Low Order Position of Header Label
53-68	Blank
69	Card Indicator
70	Tape Indicator
71	Printer Indicator
72	Not Used
73	History Indicator
74-76	Save Statement Number
77-80	Low Order Position of OEDT
81-84	SDA of Selected Output Table
85-88	SDA of Record Descriptor Table
89-92	Number Records for Record Desc. Table
93-96	SDA of Output Entry Desc. Table
97-100	Number Records for OEDT

RECORD DESCRIPTOR TABLE

Created By: LID

Purpose: Each entry describes a particular type of input record which may be encountered during generation of a Serial File Tape.

Table Size: 1 record of 2800 characters allocated for each entry in the Report Descriptor Table.

Entry Size: 40 characters per entry, recorded in Move mode.

End of Table: A record mark in the first position of an entry indicates the end of the table.

Entry Description:

<u>Positions</u>	<u>Contents</u>
1-5	Record Type
6-8	Low Order Position of Record Type
9	Length of Record Type
10	Input Media
11-12	Maximum Number Records per Entry
13	Number of Control Attributes
14	Number of Sub-Control Attributes
15-17	High Order Position of Remark Text
18-20	Low Order Position of Remark Text
21-32	Blank
33-36	SDA of Attribute Desc Table
37-40	Number Records for Attribute Desc Table

ATTRIBUTE DESCRIPTOR TABLE

Created By: LID

Purpose: Each Attribute Descriptor Table describes all of the attributes associated with a particular record type.

Table Size: 1 or more 2800 character records (sufficient to describe all attributes pertaining to a given record type).

Entry Size: 40 characters per entry, recorded in Move mode.

End of Table: A record mark in the first position of an entry indicates the end of the table.

Entry Description:

<u>Positions</u>	<u>Contents</u>
1-17	Attribute Name
18-19	Length of Attribute
20-22	Low Order Position of Attribute
23	Data Class
24	Audit Check Indicator
25	Mandatory Value Indicator
26-30	Record Type
31	Type of Attribute
32-36	Blank
37-39	Low Order Position of the Attribute Value in the Output Entry File
40	Not Used

SELECTED OUTPUT TABLE (SOT)

Created By: LID. The space is allocated by LID, but table entries are actually loaded by LOD.

Purpose: Each SOT table acts as a directory for up to seven output programs which may be associated with a given report number.

Table Size: 1 record of 245 characters (only the first 210 characters are actually used) is allocated for each entry in the Report Descriptor Table.

Entry Size: 30 characters per entry, recorded in Move mode.

End of Table: A record mark in the first position of an entry indicates the end of the table.

Entry Description:

<u>Positions</u>	<u>Contents</u>
1	Entry Indicator
2-5	SDA of Condensed Output Format Statements
6	Output Media
7-10	Output Entry Size for this Report
11	History Tape Indicator
12-14	Highest Statement Number for this Output Program
15-16	Total Output Statements Fitting into Core with this Format
17-21	Core Origin of Output Statements for this Output Format
22-26	Core Origin of Input Area for this Output Format
27	Usually Blank. Set to 1 by UPSTAT to indicate that the condensed statements to which this SOT entry points have been fully condensed by UPSTAT.
28-30	Blank

OUTPUT ENTRY DESCRIPTOR TABLE (OEDT)

Created By: LID

Purpose: This table describes the format of the records on the Serial File Tape.

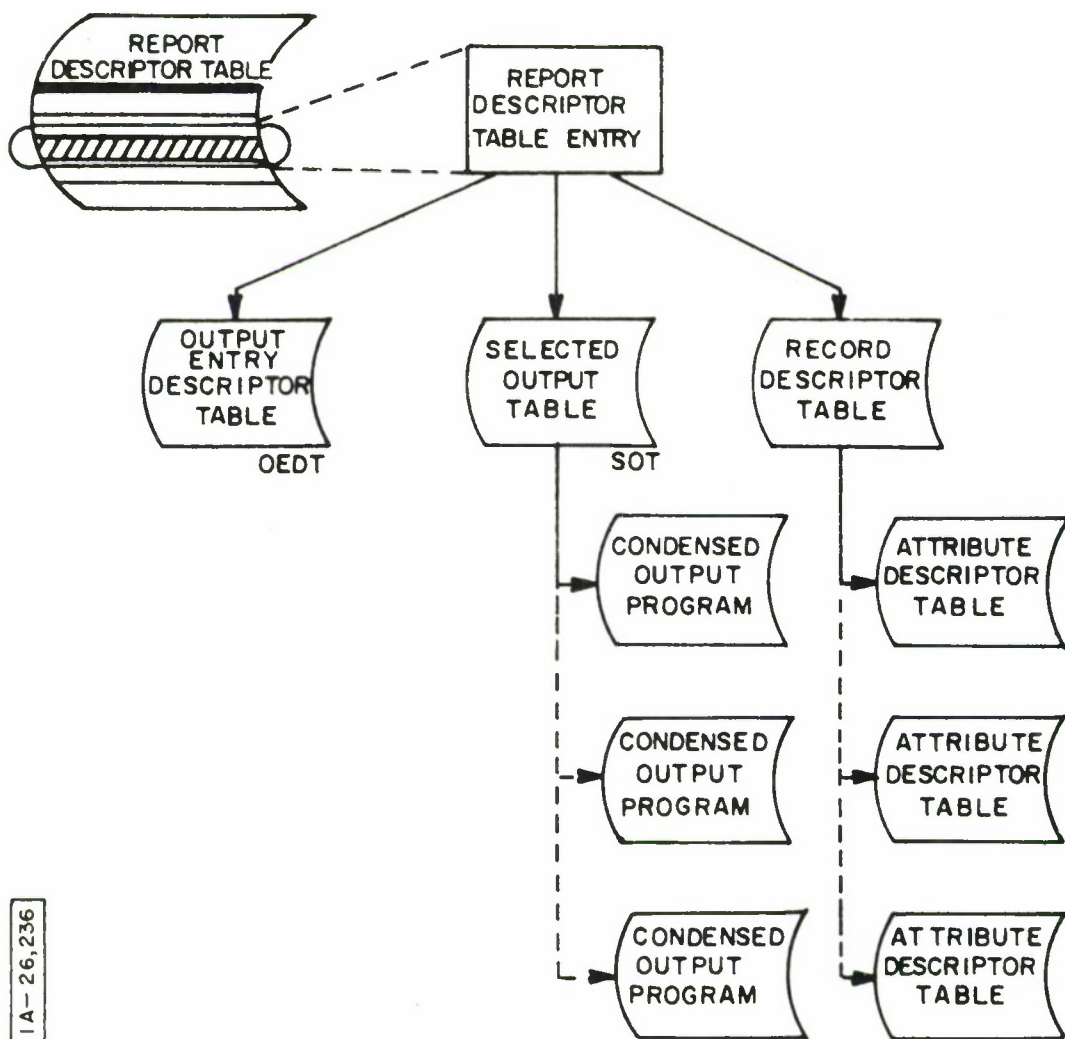
Table Size: 1 record of 2800 characters.

Entry Size: 25 characters per entry, written in Move mode.

End of Table: A record mark in the first position of an entry indicates the end of the table.

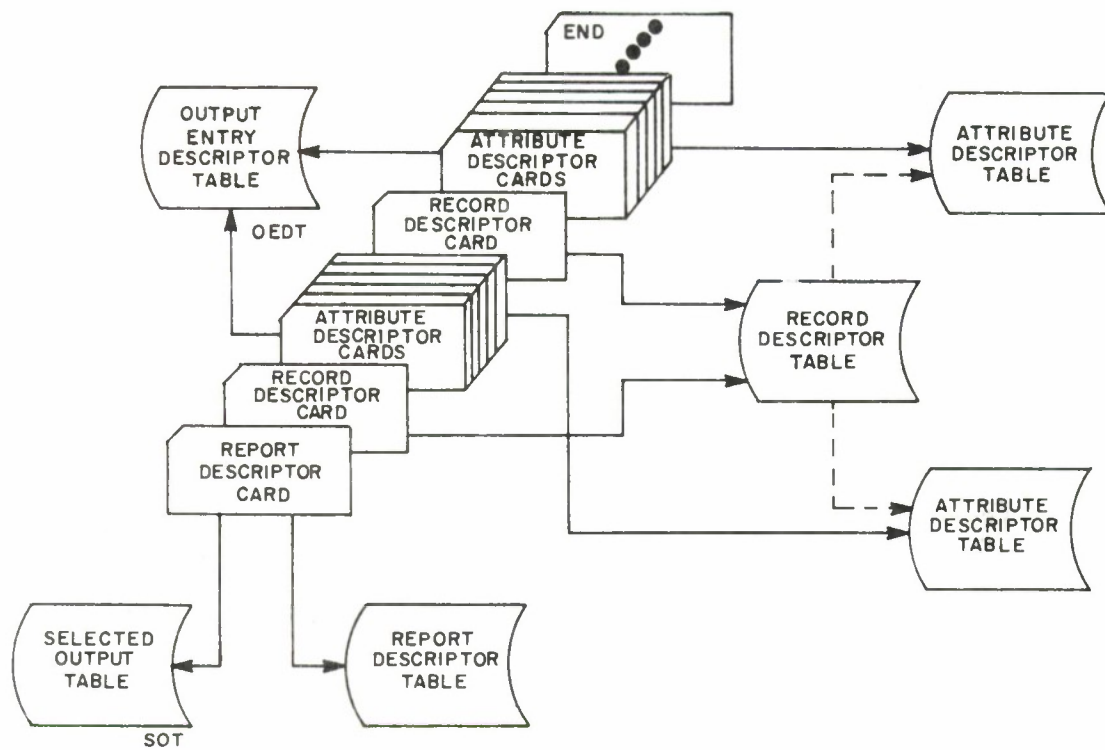
Entry Description:

<u>Positions</u>	<u>Contents</u>
1-17	Attribute Name
18	"Zero" inserted by LID
19-20	Length of Attribute
21-23	Low Order Position of Attribute
24-25	Blank - Not Used



IA-26,236

Figure 2. Tables Associated with One Report Descriptor Table Entry



IA-26,235

Figure 3. LID Input and Output, Mode 1

LEGITIMATE VALUE ATTRIBUTE NAME TABLE

Created By: LEGVAL

Purpose: This table contains the name of every attribute to be checked for legality during file generation, and acts as a directory for the lists of legal values.

Table Size: 10 records of 245 characters each.

Entry Size: 24 characters per entry, recorded in Move mode.

End of Table: A record mark in the first position of an entry indicates the end of the table.

Entry Description:

<u>Positions</u>	<u>Contents</u>
1-17	Attribute Name
18-21	SDA of Legitimate Values for this Attribute.
22	Number of records assigned to above SDA.
23-24	Maximum length of attribute value.

LEGITIMATE VALUE TABLE

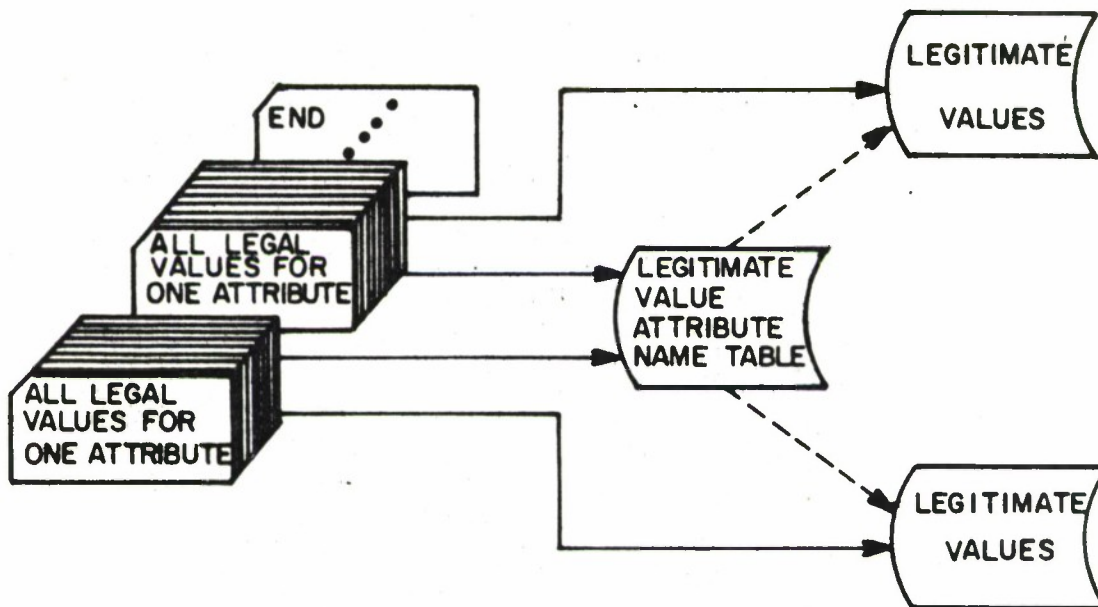
Created By: LEGVAL

Purpose: Each Legitimate Value Table contains the list of all allowable values for one particular attribute.

Table Size: 4 records of 245 characters each.

Entry Size: Variable. The entries in a given Legitimate Value Table are as long as the maximum length of that particular attribute value.

End of Table: A record mark in the first position of an entry indicates the end of the table.



IB - 26, 238

Figure 4. LEGVAL Input and Output

CONDENSED OUTPUT PROGRAM . . .

Created By: LOD

Purpose: Each table contains one Output Program, processed and condensed by LOD and available for execution.

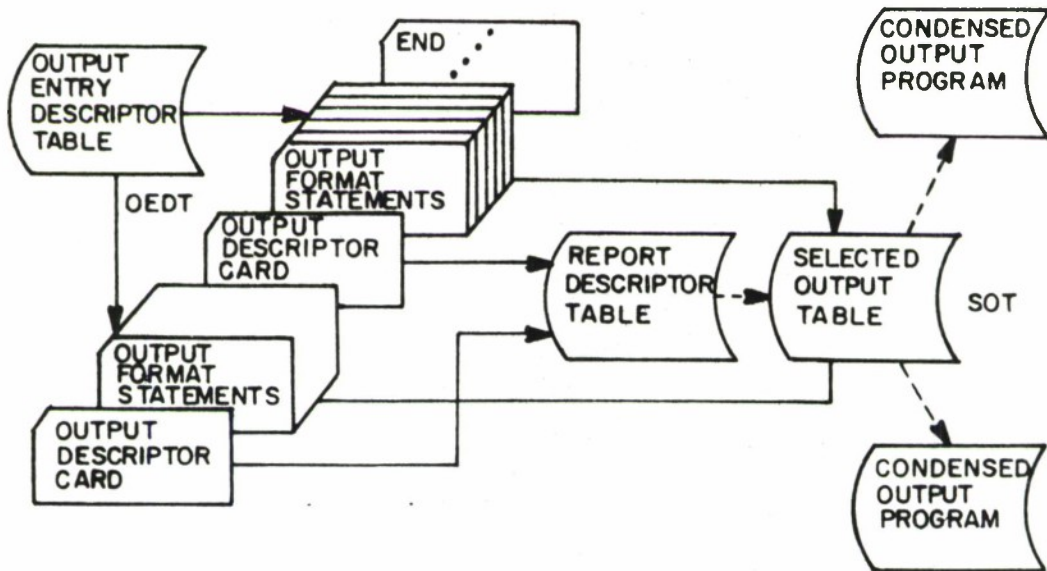
Table Size: The table contains one 245 character record for each statement of the Output Program (only the first 200 characters of each record are actually available for use).

Entry Size: Each entry (condensed statement) is contained in a 245 character record written in Move mode.

End of Table: Each condensed statement is assigned within an SDA to the relative record corresponding to the statement number. Thus, the last entry in the table is contained in the relative record corresponding to the highest statement number used in the Output Program.

Entry Description:

See program description of LOD.



IB - 26, 239

Figure 5. LOD Input and Output

SAVED STATEMENT TABLE DIRECTORY

Created By: SAVER

Purpose: This table serves as a directory for locating the Query Language statements which have been stored on the system disk by SAVER.

Table Size: 1 record of 2800 characters (only the first 2000 are used).

Entry Size: Each entry is 4 characters in length.

End of Table: The end of the table is indicated by the 500th entry or a blank entry, whichever occurs first.

Entry Description:

Each entry is an SDA of a saved Query Language statement. The relative position of the entry in the table corresponds to the statement number previously assigned to the saved statement.

SAVED STATEMENT TABLE

Created By: SAVER

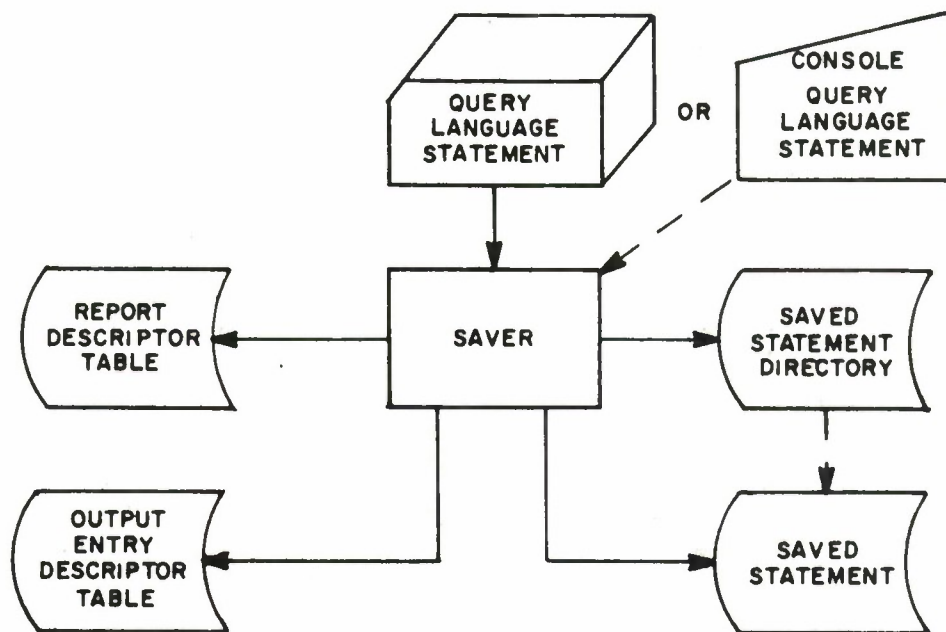
Purpose: Query Language retrieval statements are stored on the system disk for subsequent recall and execution.

Table Size: A maximum of 500 2800 character records.

Entry Size: Each saved statement is contained in one 2800 character record.

Entry Description:

See program description for SAVER.



1A-26,234

Figure 6. SAVER Input and Output

I/O COMMUNICATION RECORD

Created By: REPGEN (or by RGPBYP when using bypass mode).

Purpose: This record contains information which must be passed from the input phase to the output phase.

Table Size: 1 record of 245 characters.

Entry Size: 1 entry of 126 characters.

Entry Description:

<u>Positions</u>	<u>Contents</u>
1-100	Report Descriptor Table entry for report being processed.
101-117	AS OF TIME entered on console
118-120	High order position of Remarks text
121-123	Low order position of Remarks text
124	Format number of requested Output Program
125	Classification code of requested report
126	Number of additional copies requested

SECTION IV

OUTPUT PHASE COMMUNICATION ZONE

<u>Name</u>	<u>Loc</u>	<u>Length</u>	<u>Tag</u>	<u>Meaning</u>
WAREA	2500	200	High	Area from which a condensed statement is executed.
ENTIND	2700	1	Low	Indicator for entry from SOT to be used. A or C = active; D or E or 0 = inactive.
ENDSDA	2704	4	Low	SDA of condensed statements for this Output Program.
ENTMED	2705	1	Low	Output device for this Output Program. C = card punch, T = tape, P = printer.
ENTSIZ	2709	4	Low	Size of the output image for this Output Program.
ENTHIS	2710	1	Low	History tape indicator. Y = yes, 0 = no.
ENTHIH	2713	3	Low	Highest statement number for this condensed Output Program.
ENTAMT	2715	2	Low	Number of condensed statements that can fit in core for this Output Program.
ENTSTA	2720	5	Low	Core Origin of the condensed statements for this Output Program.
ENTINP	2725	5	Low	Core origin of the input area for this Output Program.
ENTRY	2729	4	Low	The low-order character of the SOT entry in use.
SOT	2730	245	High	Selected Output Table for this report.
OVFLOW	2976	2	Low	The line after which overflow is to occur.
COUNTR	2977	51 x 17	High	The 50 Counters available to an Output Program, plus the REMAINDER Counter (Counter 0).

<u>Name</u>	<u>Loc</u>	<u>Length</u>	<u>Tag</u>	<u>Meaning</u>
ASF DAT	3860	17	Low	Location where AS OF TIME from console is stored.
DATE	3877	17	Low	Location where system date is stored (referred to by system attribute name TODAY).
CLASS	3894	17	Low	Report classification.
SWITCH	3895	10	High	The 9 Switches available to an Output Program.
LNCTER	3906	2	Low	Line counter.
COM1NO	3909	3	Low	The statement number of the first level statement in operation.
COM1PO	3914	5	Low	The position in the first level statement in operation.
COM1AC	3915	1	Low	First level active indicator; 1 = active; 0 = inactive.
COM2NO	3918	3	Low	The statement number of the second level statement in operation.
COM2PO	3923	5	Low	The position in the second level phrase in operation.
COM2AC	3924	1	Low	Second level active indicator; 1 = active; 0 = inactive.
COMREP	3927	3	Low	The statement number of the REPEAT statement in operation.
COMSOR	3932	5	Low	The core origin of the condensed statements.
COMSNO	3934	2	Low	The number of statements in core.
COMIOR	3939	5	Low	The core origin of the input area.
COMIEN	3944	5	Low	The end of the input area in core.
COMOOR	3949	5	Low	The core origin of the output area.

<u>Name</u>	<u>Loc</u>	<u>Length</u>	<u>Tag</u>	<u>Meaning</u>
COMOEN	3954	5	Low	The end of the output area in core.
COMTES	3955	1	Low	The result of the last IF test; 1 = true; Ø = false.
COMPUN	3956	1	Low	The punctuation from the last phrase passed.
CONDON	3957	1	Low	Finished indicator; 1 = no more input data.
COMSTK	3958	1	Low	The stacker into which the next card to be punched will be placed.
COMQL	3959	1	Low	Query Language indicator. Non blank = Query Language.
COMTYP	3960	1	Low	The statement type of the last statement operated.
COMDR	3961	1	Low	Type of first-level statement last executed; D = DO; R = REPEAT.
COMOVF	3962	1	Low	Overflow indicator.
HILO	3968	6	Low	The high and low order addresses of remarks data on remarks card.
COMNEW	3969	1	Low	Indicator that a new statement has become active.
COPIES	3970	1	Low	Number of additional copies of the report requested.
NEWPAG	3971	1	Low	Indicator that a SKIP TO NEW PAGE was executed.
SPACES	3973	2	Low	Number of blank lines printed by the last SKIP statement.
WRTSCT	3975	2	Low	Not used.

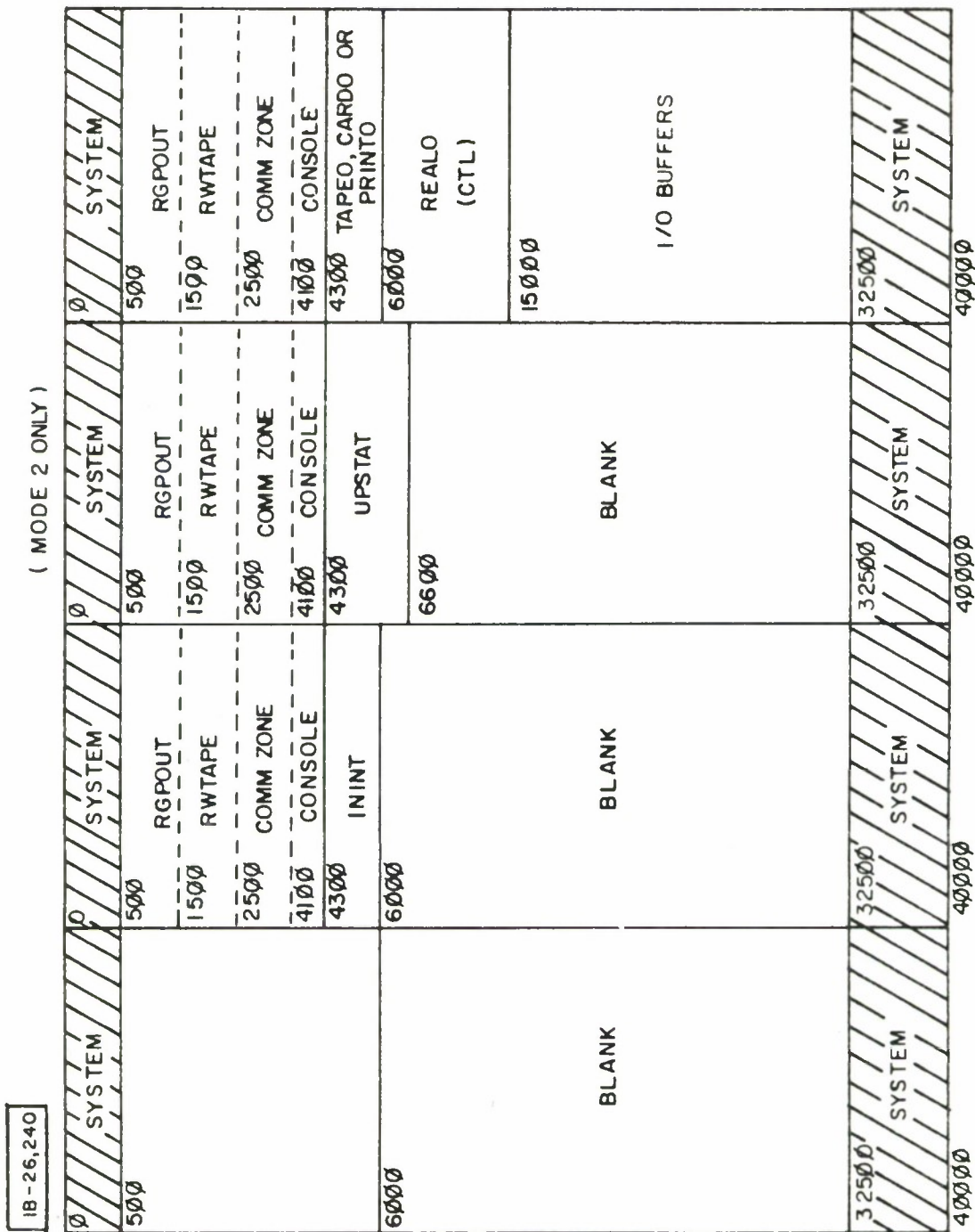


Figure 7. REPGEN Output Phase Core Map

SECTION V

SYMBOLIC DISK ADDRESSES

The REPGEN programs make frequent use of the system disk. All disk references are made through the use of DSKACC, using Symbolic Disk Addresses (SDA's). Several programs in REPGEN use the system allocation routine DSKALL to dynamically obtain needed SDA's. However, there are a number of SDA's which must be obtained prior to installing the REPGEN capability. These SDA's are assembled directly into some of the REPGEN programs.

Table 1 which follows describes the characteristics of the disk areas for which SDA's must be obtained prior to installing REPGEN.

TABLE 1

SDA'S REQUIRED BY REGEN PROGRAMS

SDA Explanation	Mode Code	# of Records	# Char. Per Rec.	USED BY																		
				LID	LOD	LEGAL	LITD	TAPETM	REGEN	RCBPYP	RCPOUT	REALO	CARDO	TAPEO	PRINTO	ININT	UPSTAT	REMARK	CHCLOD	SAVER	REKALL	DELETE
General Tape Descriptor Table	3/01		2165				X	X														
Typewriter Input Area	4/1440		2800					X	X	X	X							X	X		X	X
Current Date	8/01		245					X	X							X						
Scratch Area	4/8429		2800					X	X								X					
I/O Communication Record	8/01		245						X	X						X						
LEGAL Attribute Name Table	8/10		245			X			X													
Report Descriptor Table	4/01		2800	X	X				X	X								X	X			
Communications Zone	5/04		400								X					X						
CARDO Routine	3/01		2165								X											
TAPEO Routine	5/04		400								X											
PRINTO Routine	5/05		400								X											
REALO Routine	3/04		2165								X											
ININT Routine	5/04		400								X											
UPSTAT Routine	5/06		400								X											
REMARK Routine	3/02		2165								X											
Temporary Header Storage	6/02		529											X								
History Tape Overlay Area	5/05		400										X	X								
Table of Saved Statement SDA's	4/01		2800															X	X	X	X	X

SECTION VI
REPGEN FLOWCHARTS AND ABSTRACTS

6.1 LID

Purpose

LID reads a deck of input descriptor cards and generates a set of tables which are required by subsequent REPGEN programs.

Calling Sequence

Enter LID on the console typewriter. Return will always be to IDLOOP.

Description

There are four acceptable input card types to LID. They are the Report Descriptor Card, the Record Descriptor Card, the Attribute Descriptor Card and the END Card.

Report Descriptor Card -- When LID reads a Report Descriptor Card it does the following:

- a. Checks the Report Descriptor Table to make sure there is an open slot or that this is not a duplicate of another report number.
- b. If the input medium declared on this card is not disk, an SDA is allocated for a Record Descriptor Table.
- c. An SDA is allocated for an SOT.
- d. An SDA is allocated for an OEDT.

Record Descriptor Card -- When LID encounters a Record Descriptor Card it:

- a. Allocates an SDA for an Attribute Descriptor Table.

Attribute Descriptor Card -- An Attribute Descriptor Card causes LID to:

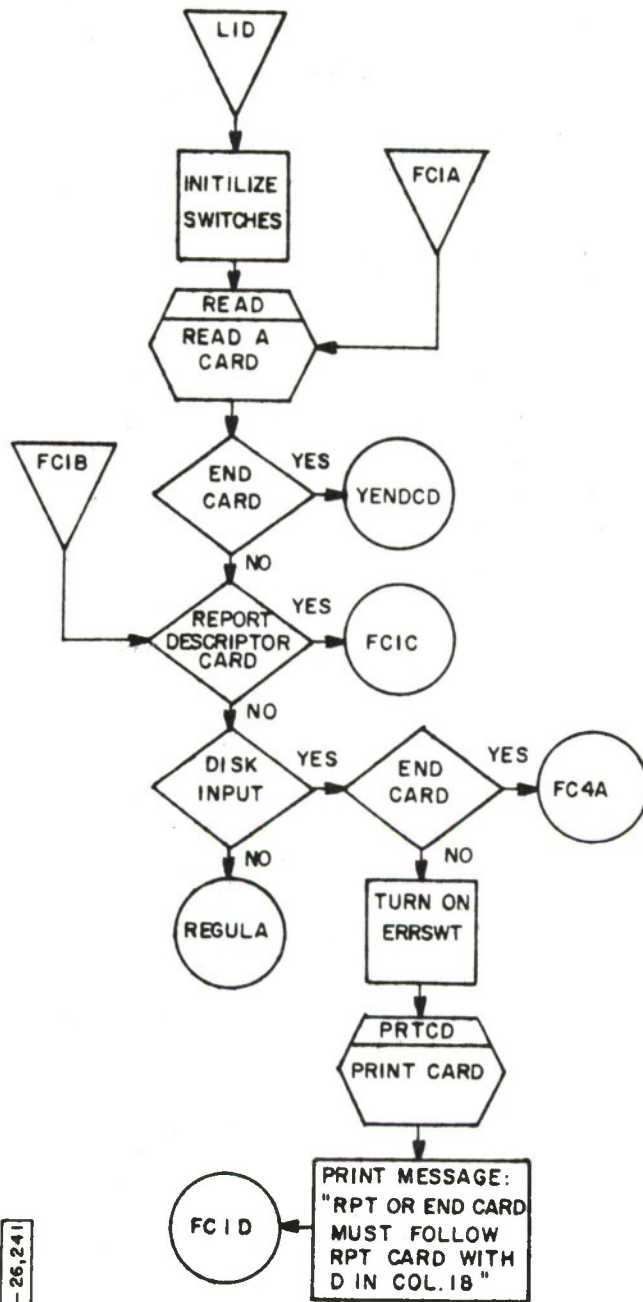
- a. Make an entry in the current Attribute Descriptor Table.
- b. Read the next Attribute Descriptor Card and repeat the process until all attributes for this record type have been treated.
- c. Make an entry in the Record Descriptor Table.

END Card (or next Report Descriptor Card)-- When an END Card is read, LID will:

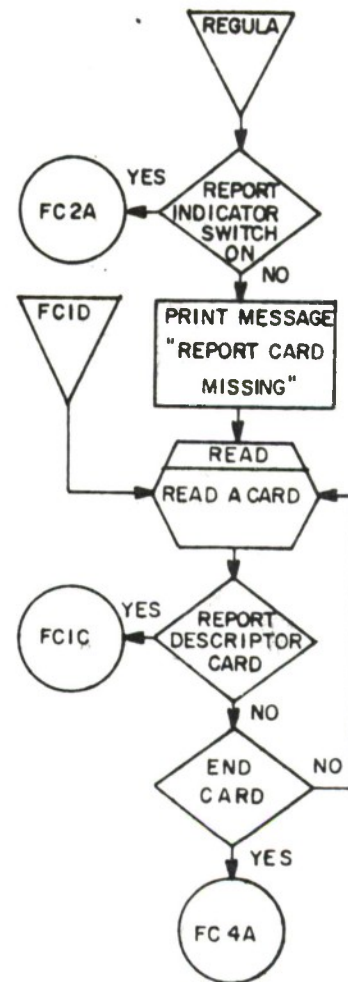
a. Construct an Output Entry Descriptor Table (OEDT) to describe the format of each record as it will later appear on the Serial File Tape.

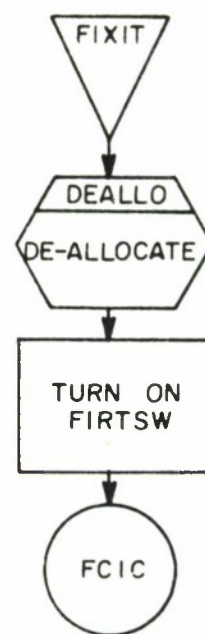
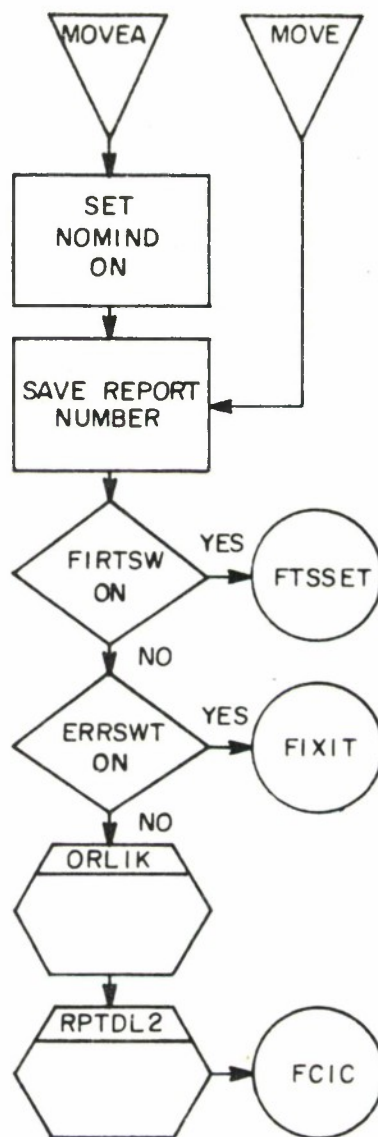
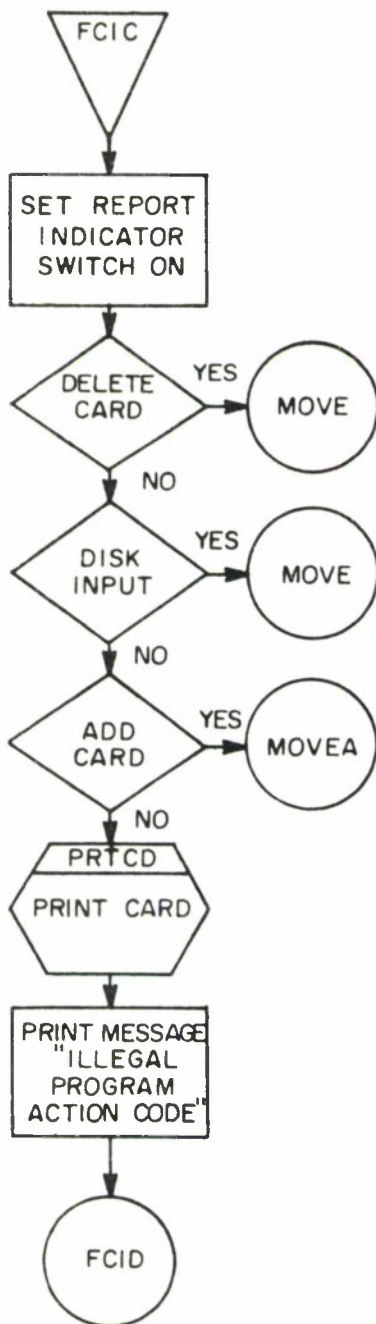
b. Modify each Attribute Descriptor Table entry to include data identifying the attribute's assigned position in the Serial File Tape records.

c. The modified Report Descriptor Table (and Record Descriptor Table if input is not disk) is written onto the system disk.

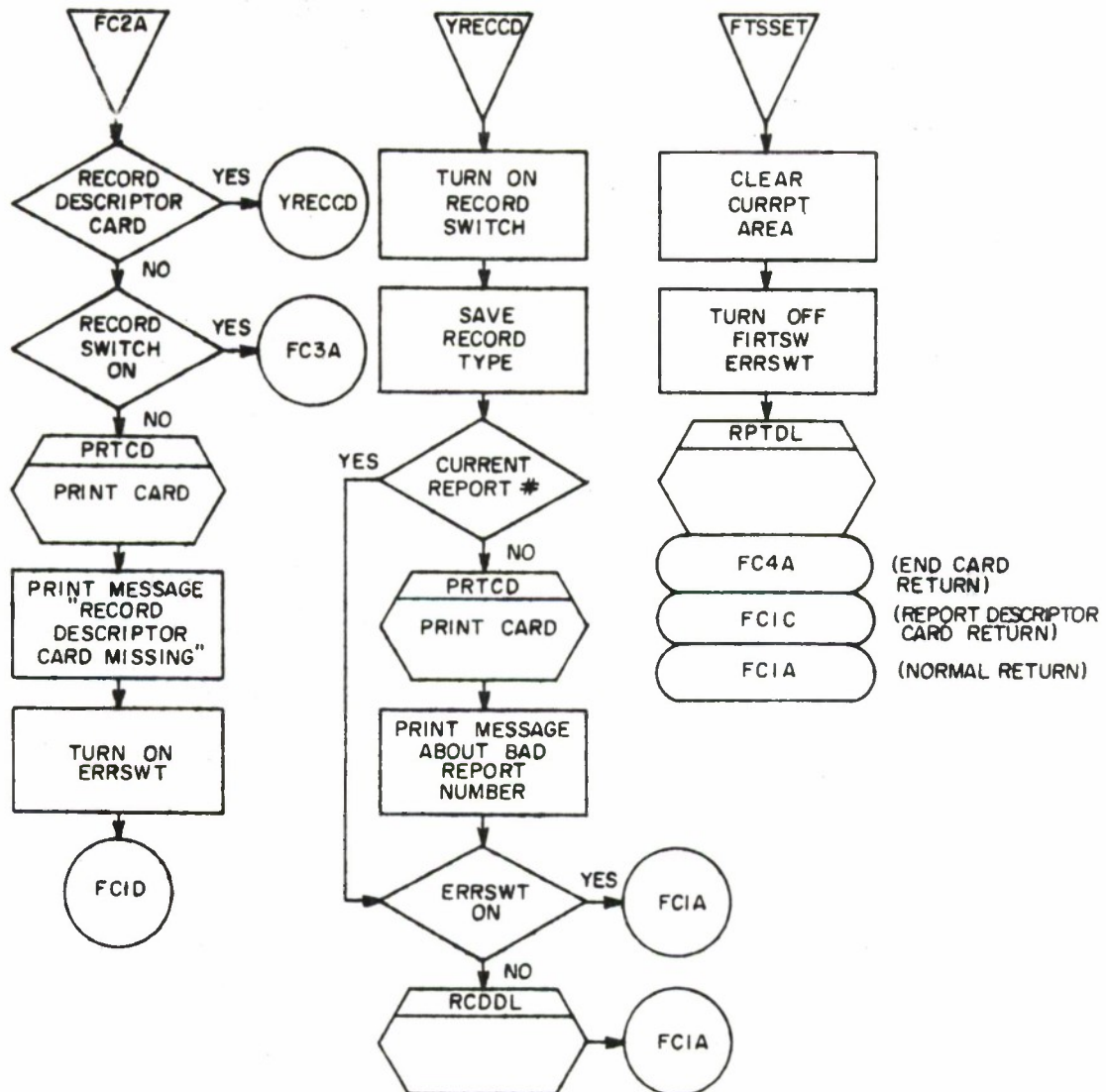


1B-26,241

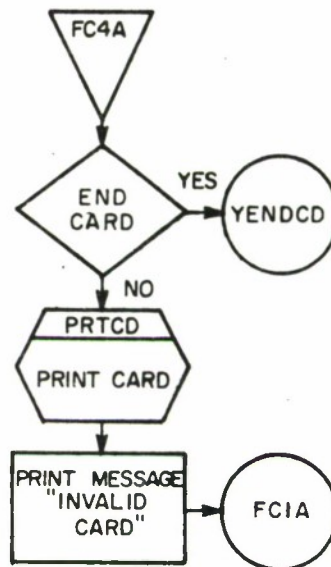
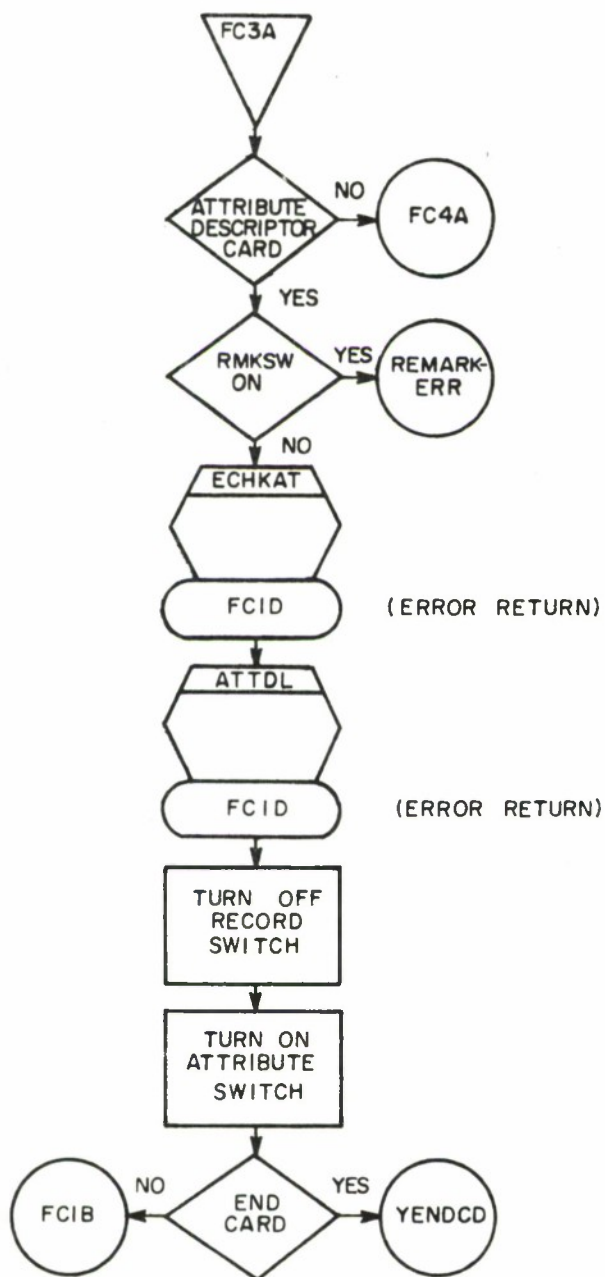




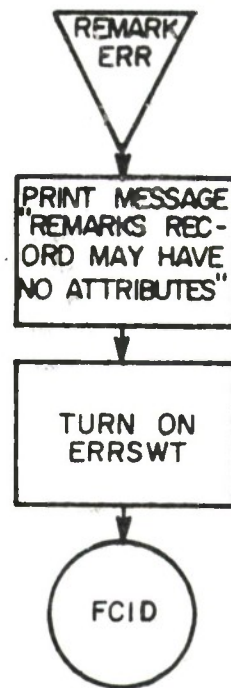
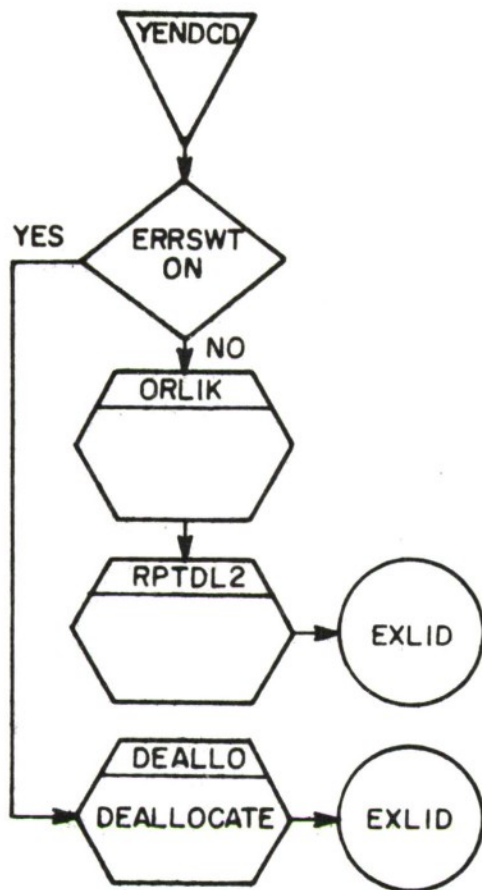
18-26,229



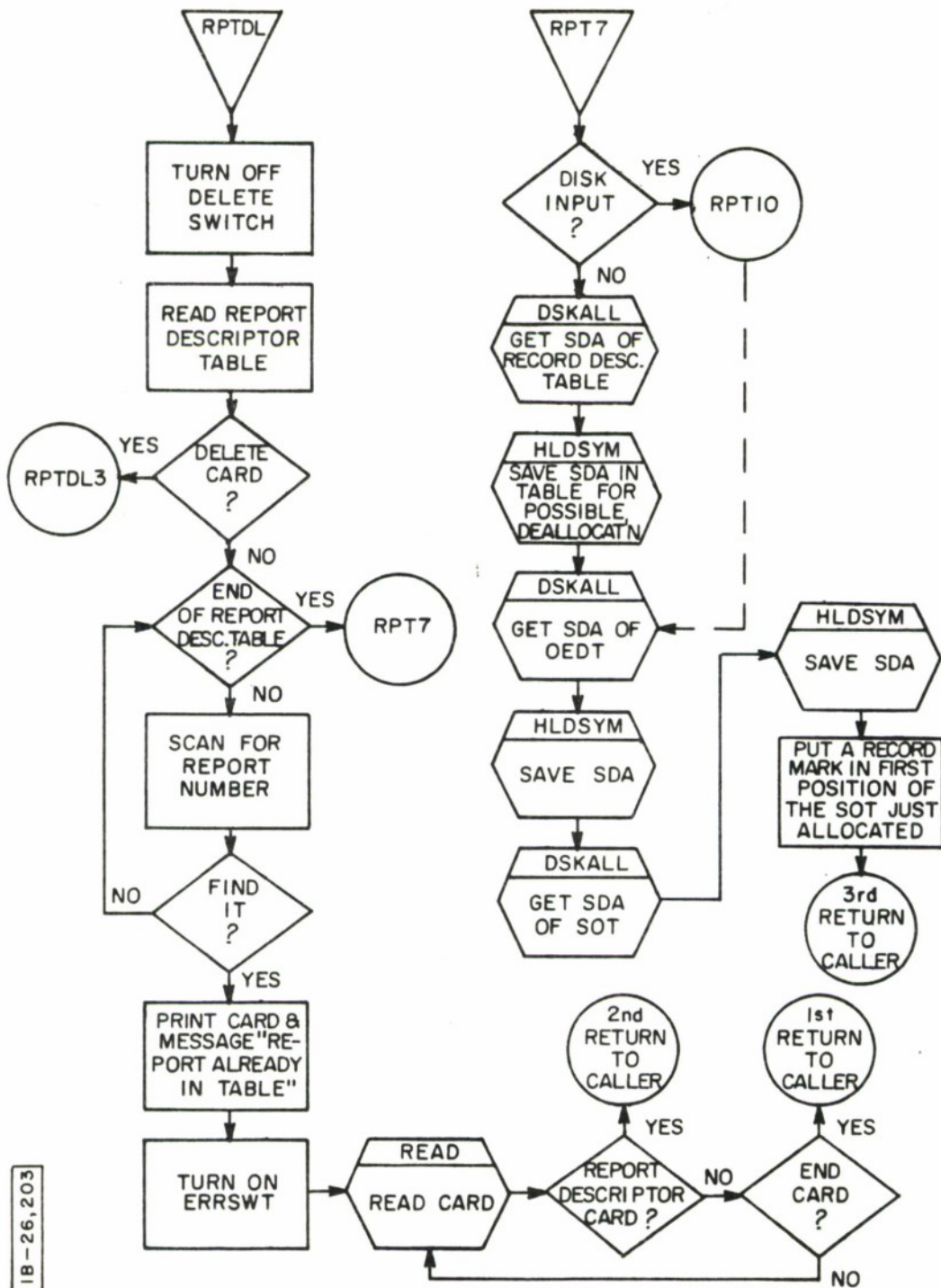
18-26,230

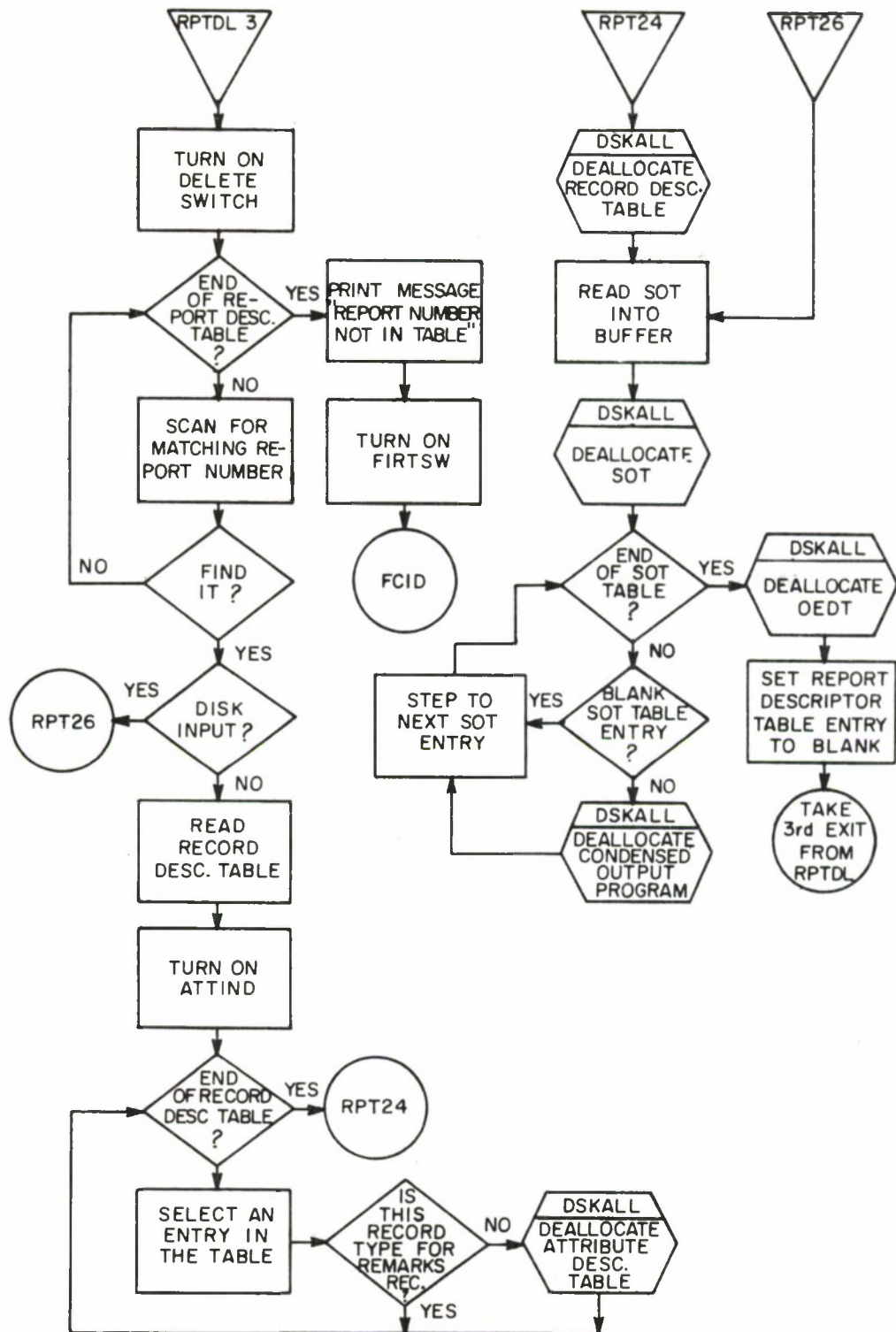


IB - 26,209

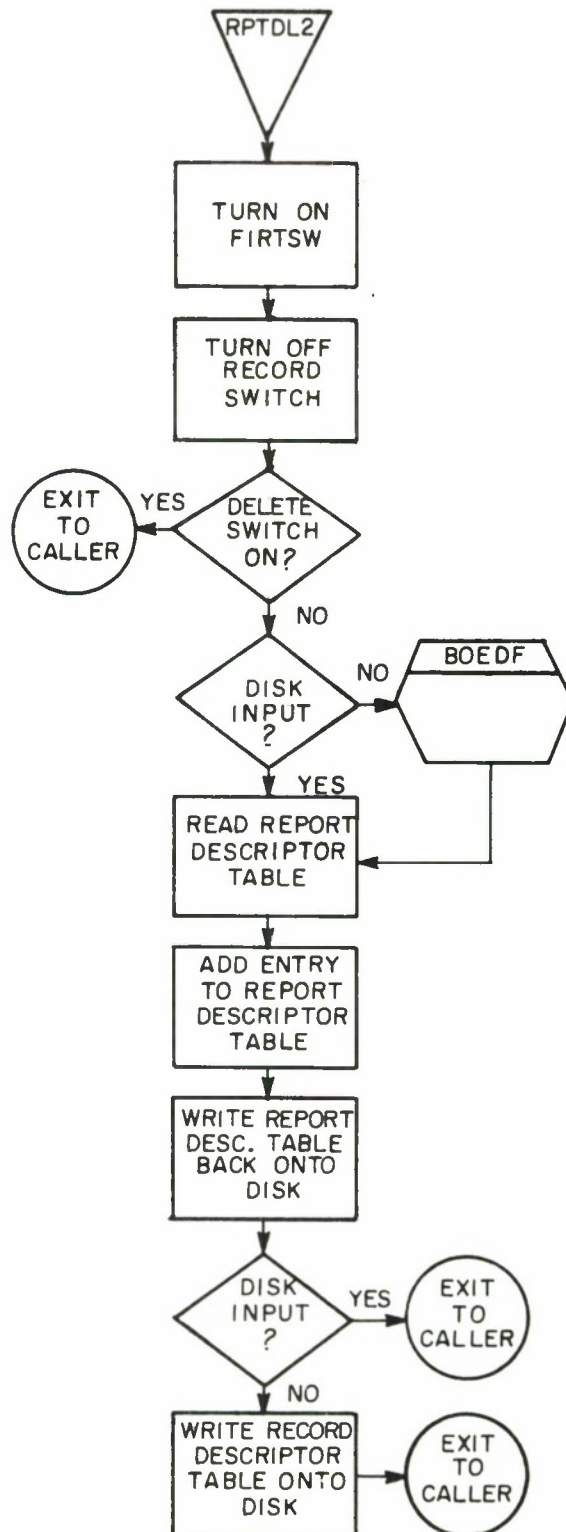


1B-26,208

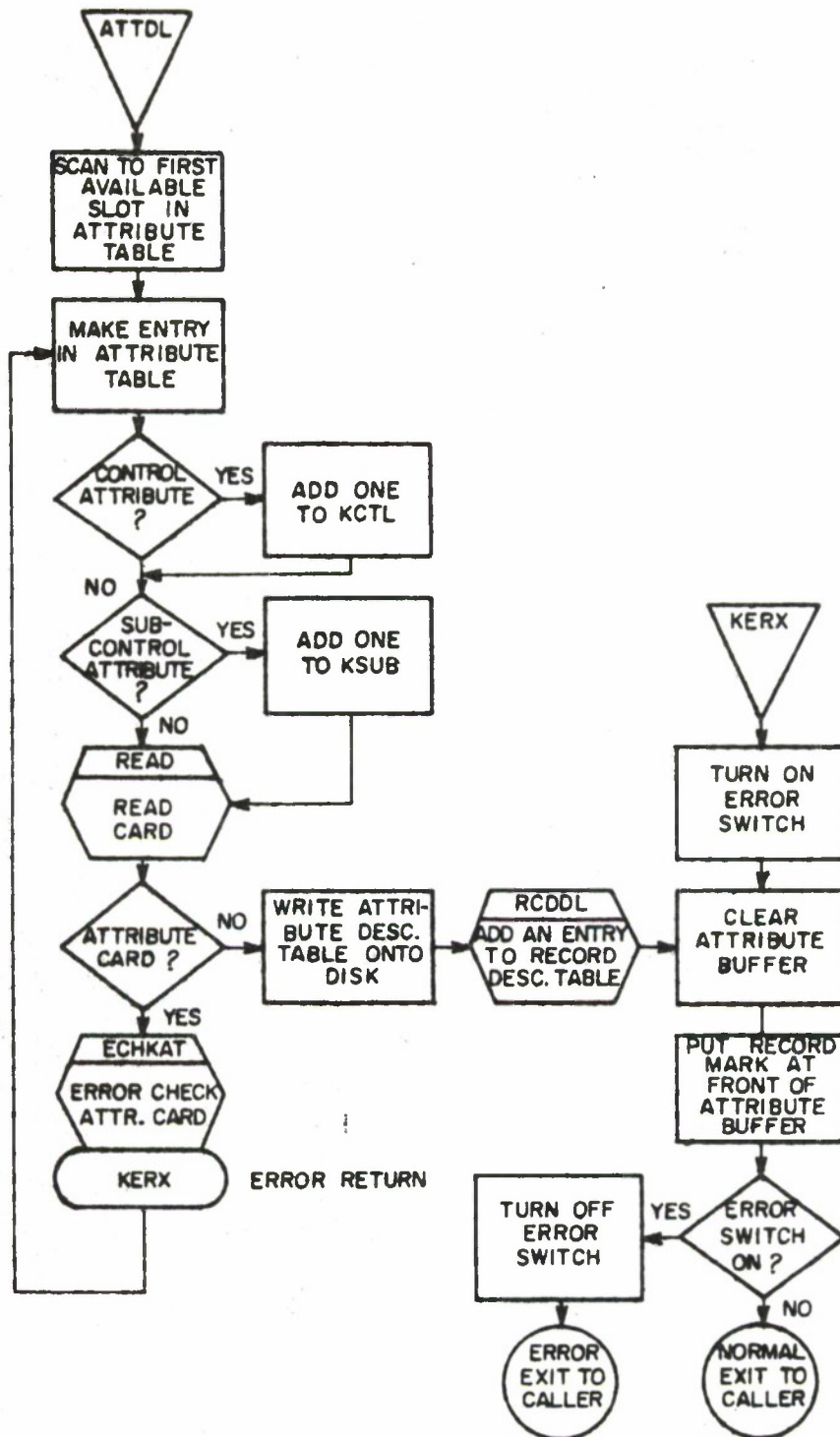




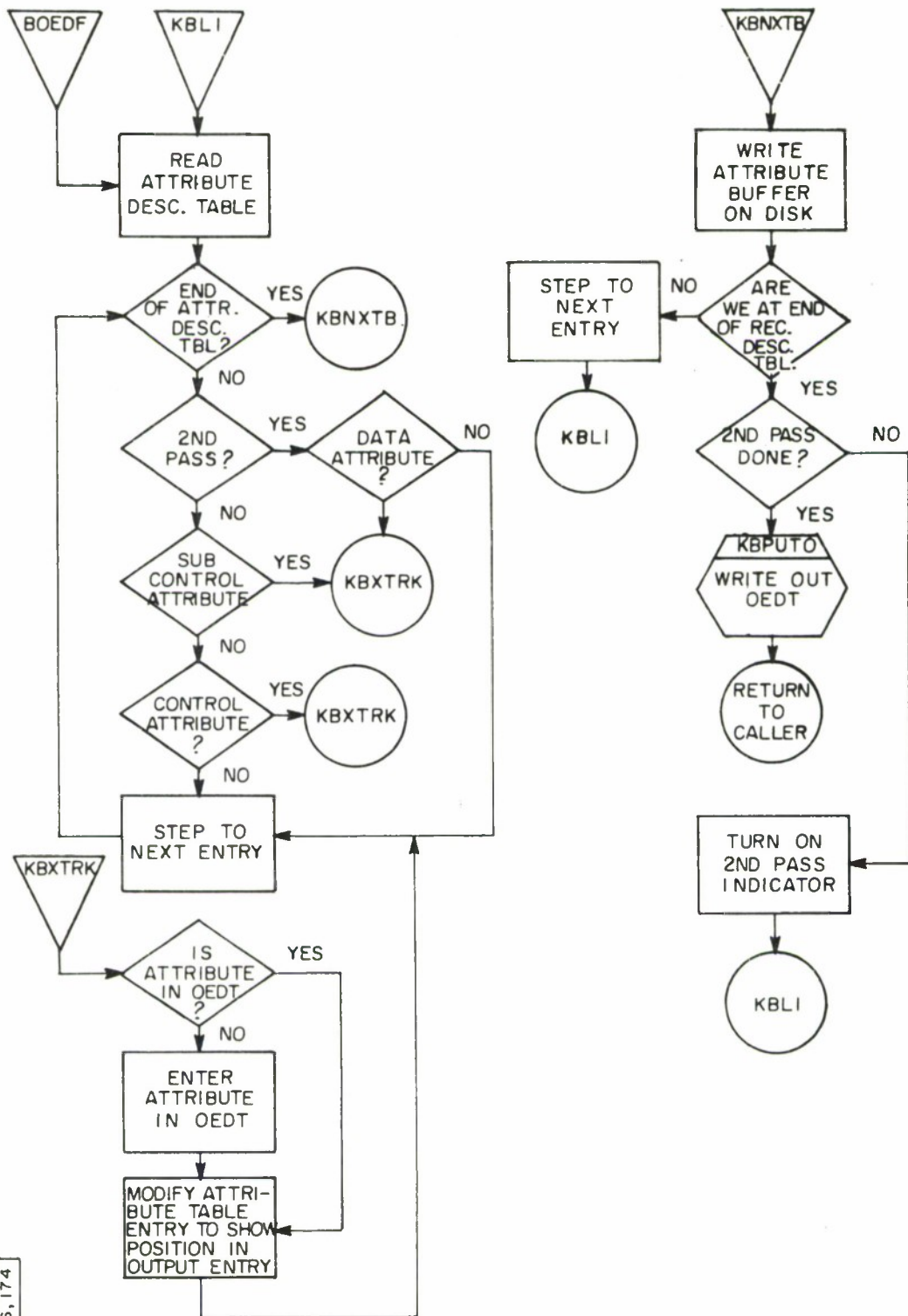
IB - 26,204



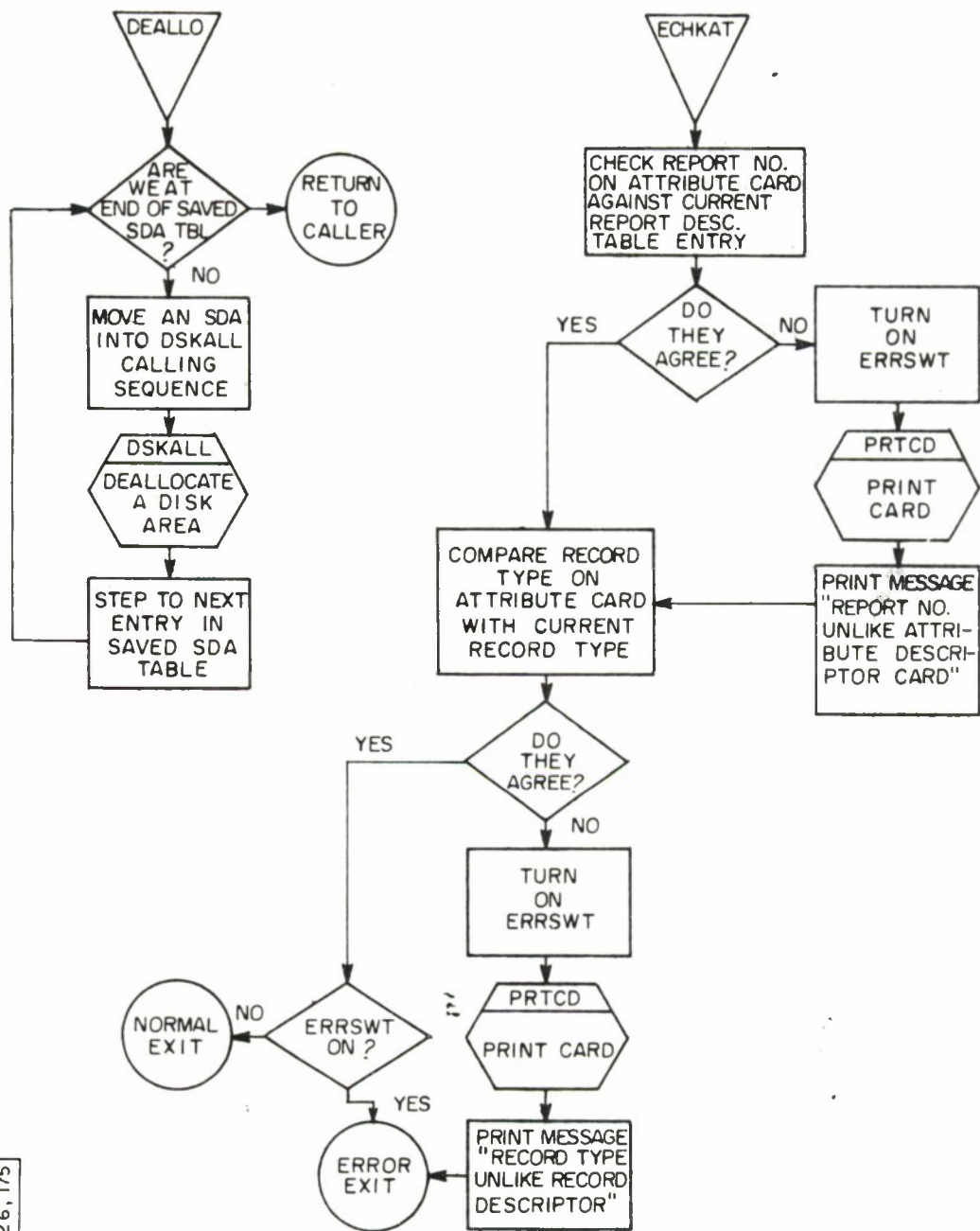
1B - 26,206



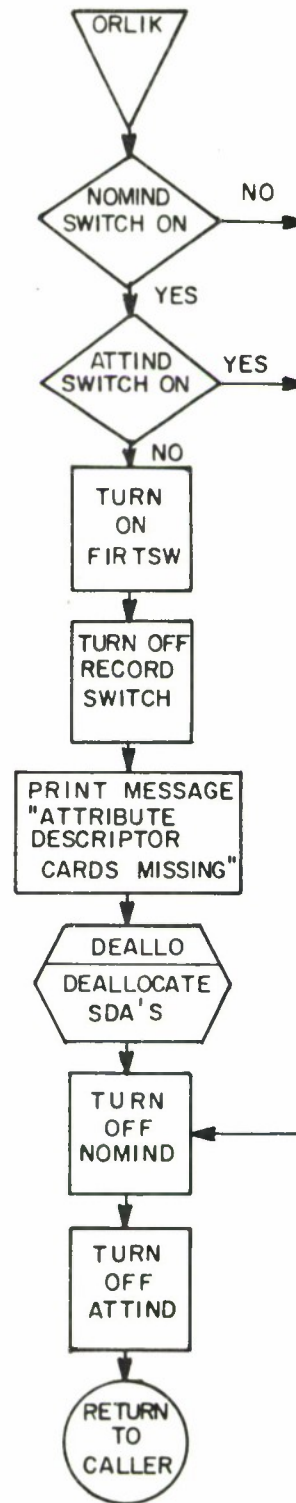
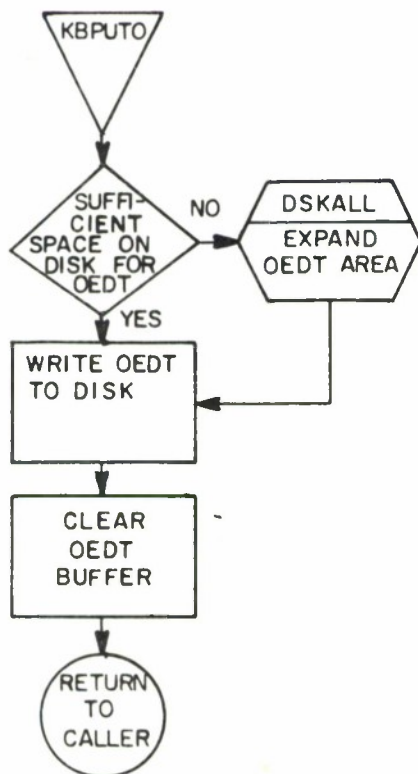
18-26,207



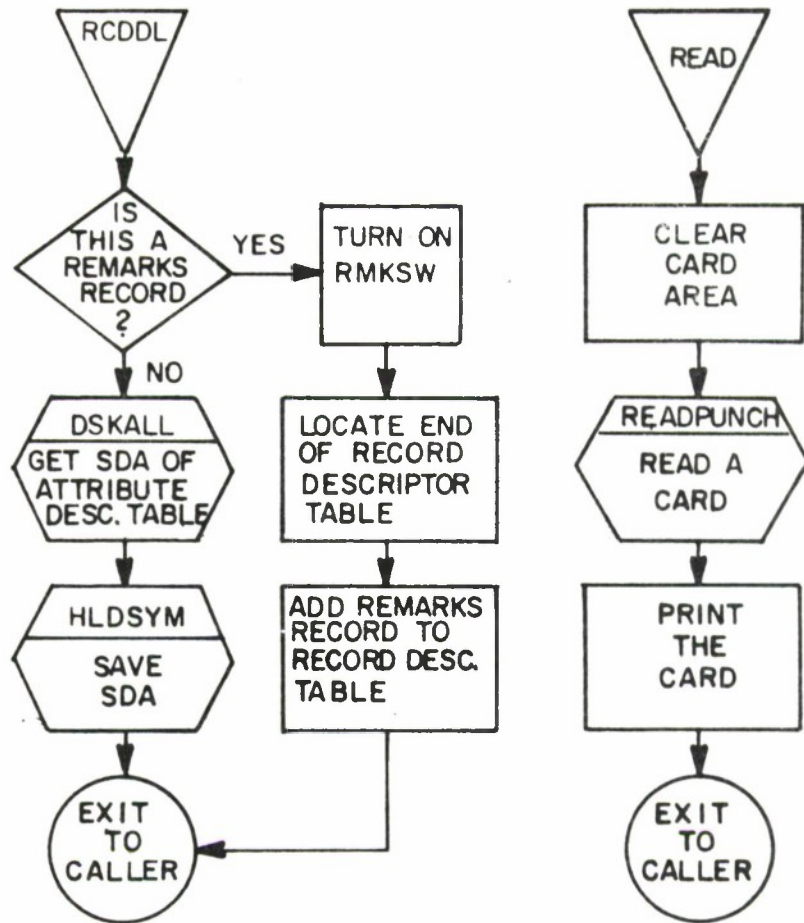
IB-26,174



IB - 26, 175



IB-26,242



IB-26,243

LEGVAL

Purpose

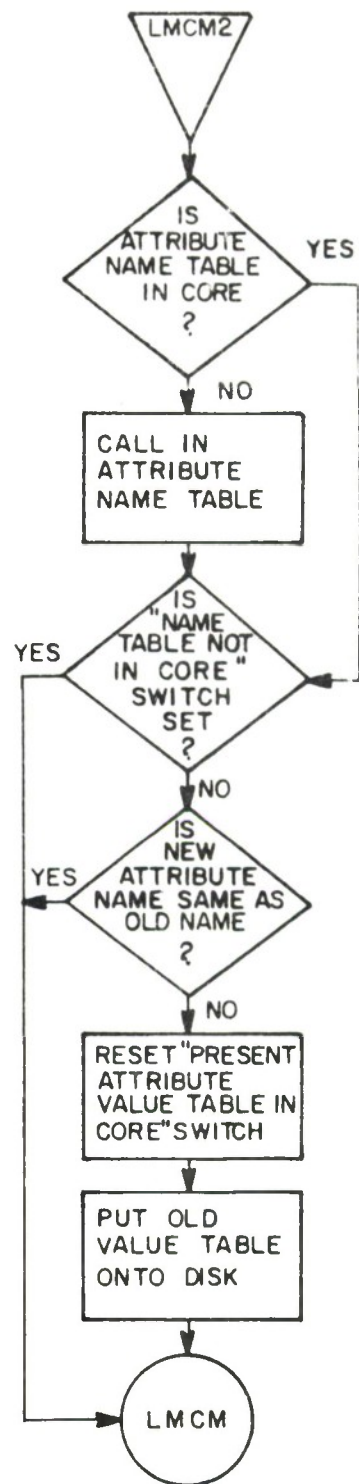
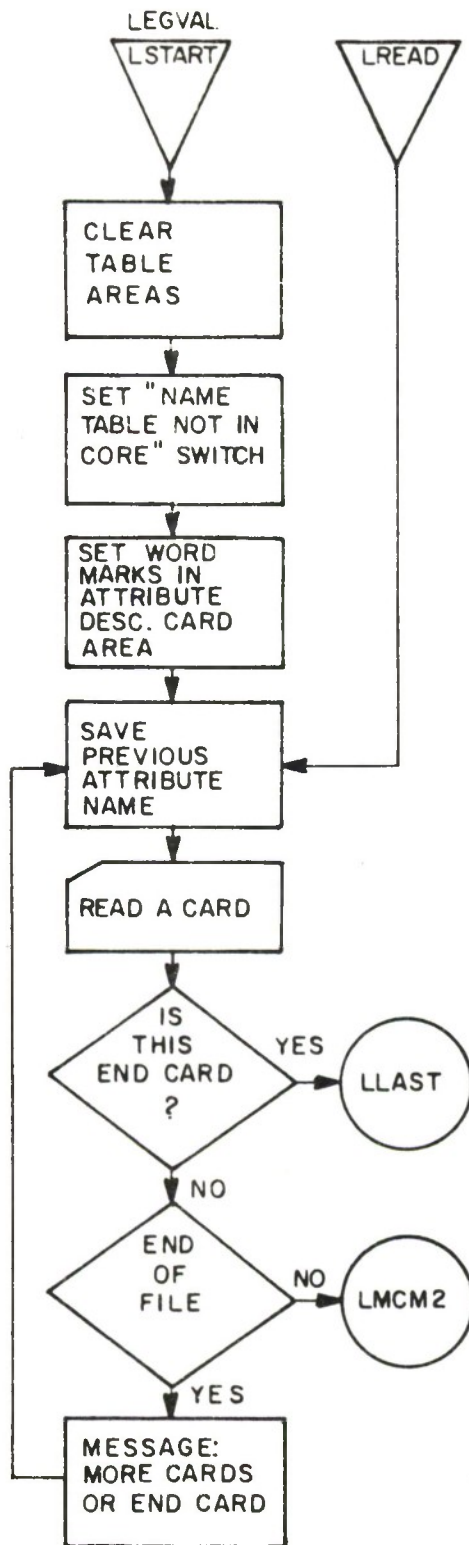
To generate a disk table containing all allowable values for one or more attributes. This table is to be used during the file generation process which creates a Serial File Tape from source input data.

Calling Sequence

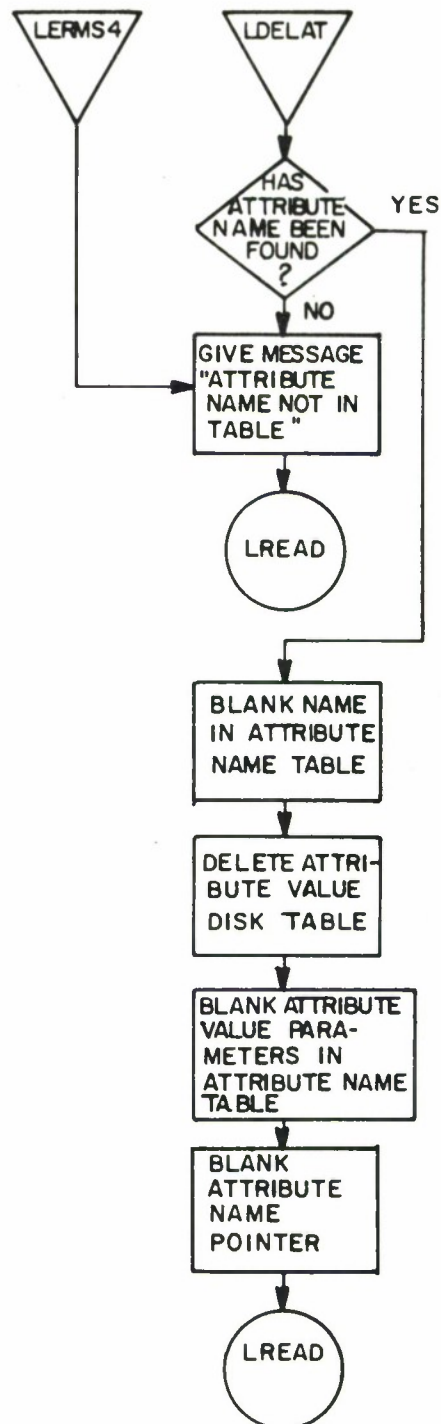
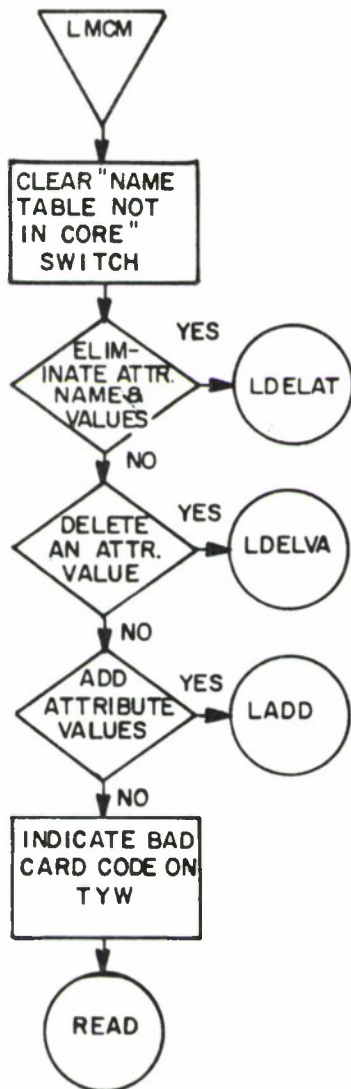
Enter LEGVAL on the console typewriter.

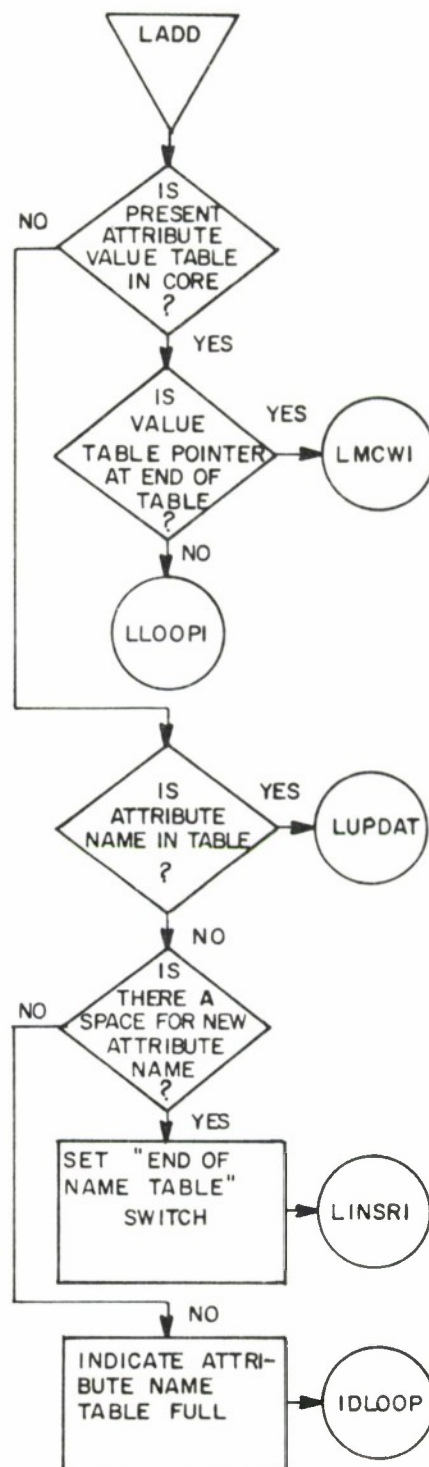
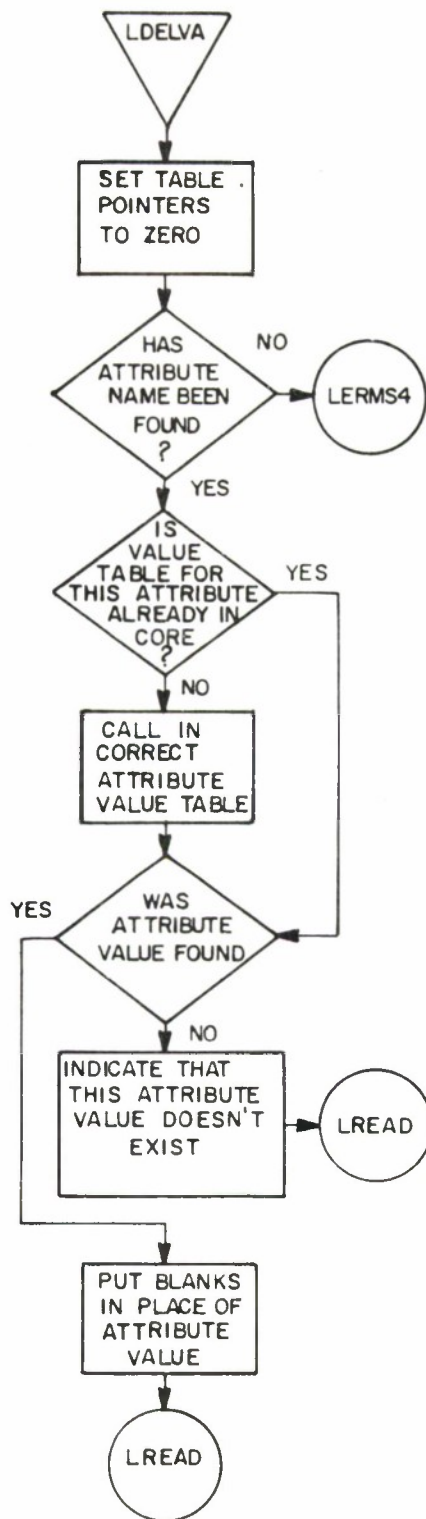
Description

Each input card to LEGVAL contains the name of an attribute and one of the allowable values for that attribute. LEGVAL checks to see if that attribute name has been previously encountered. If not, the name is added to a list of attributes which are to be legality checked. After the name has been added to the name list, an SDA is allocated to be used for storing the list of legal values pertaining to the attribute. When all allowable values for a given attribute have been read, the value table is written on the system disk. When an END card is read, the name list is written on the system disk.

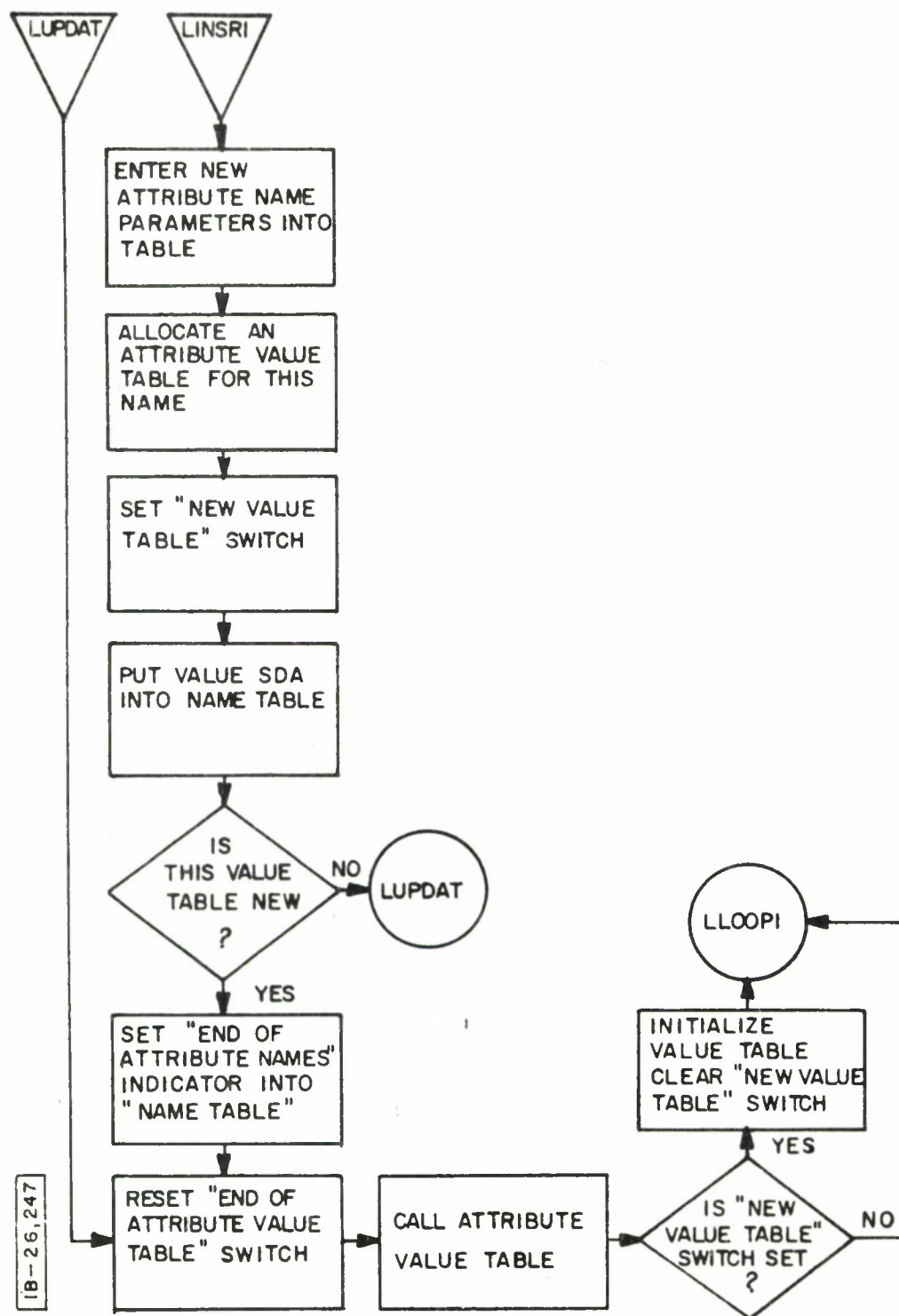


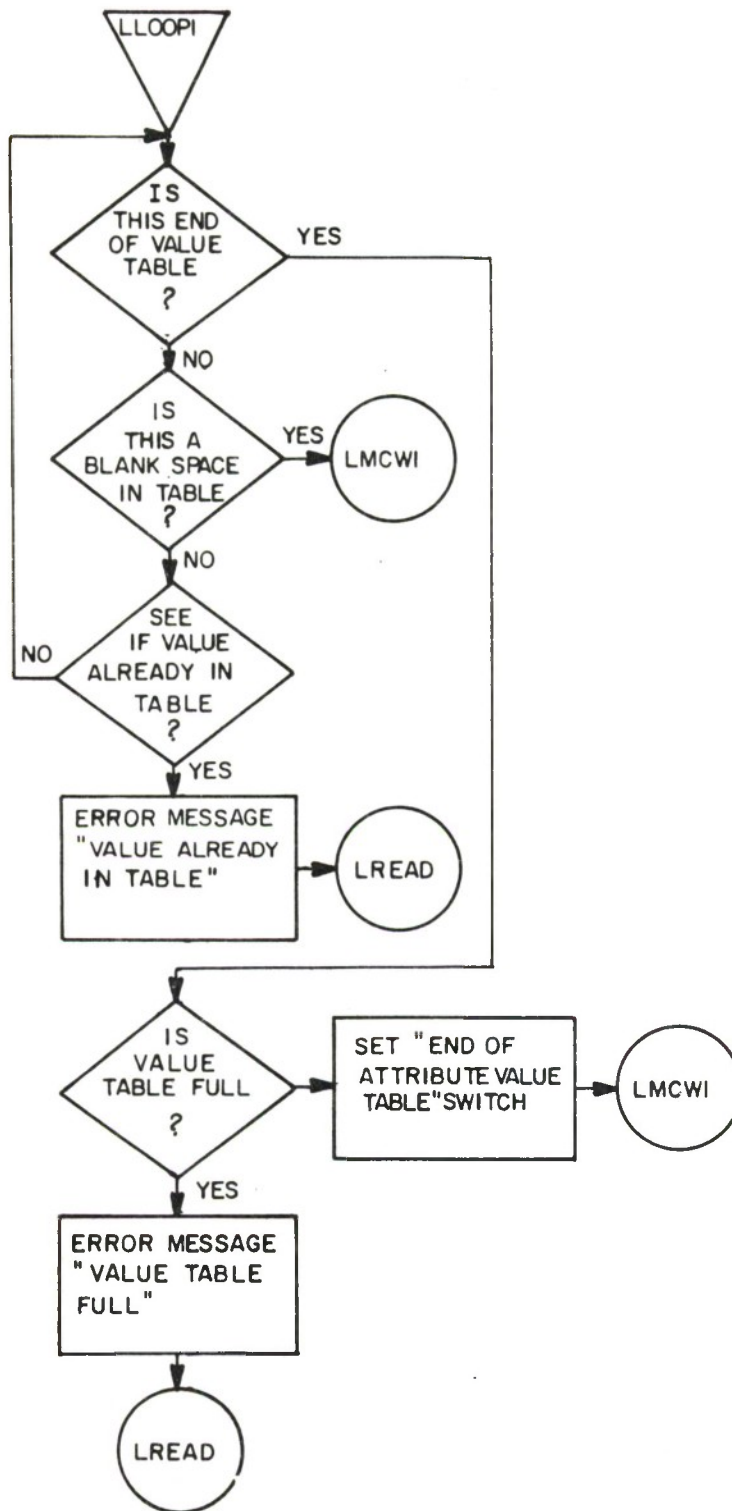
IB - 26,244



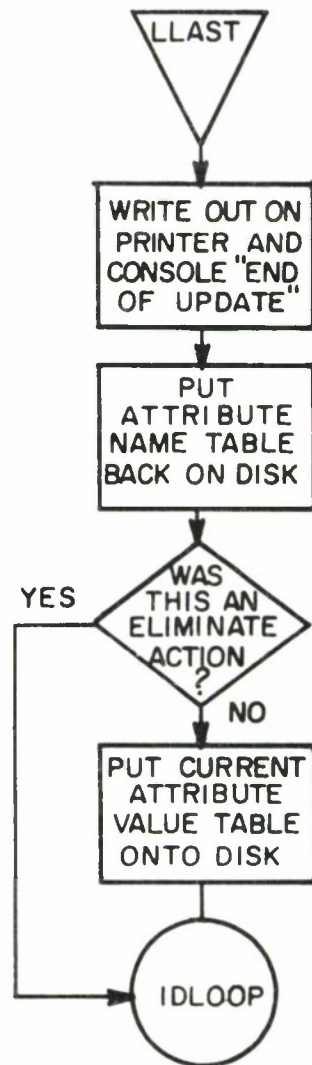
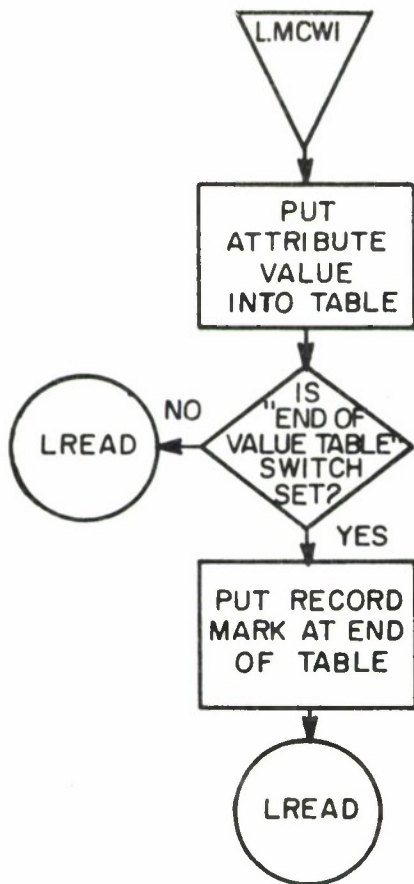


IB - 26,246





IB-26,248



18-26,192

SAVER

Purpose

A query language statement containing the GETTAC process director and report number(s) is stored on disk for utilization by the query portion of the report generation program.

Calling Sequence

From I/O typewriter, or cards with sense switch B on, starting in column 1.

```
SAVERGETTACAAAA''AA[BBB/BBB''']  
XXX+XXX+'''+XXX
```

AA''' = qualifying attributes

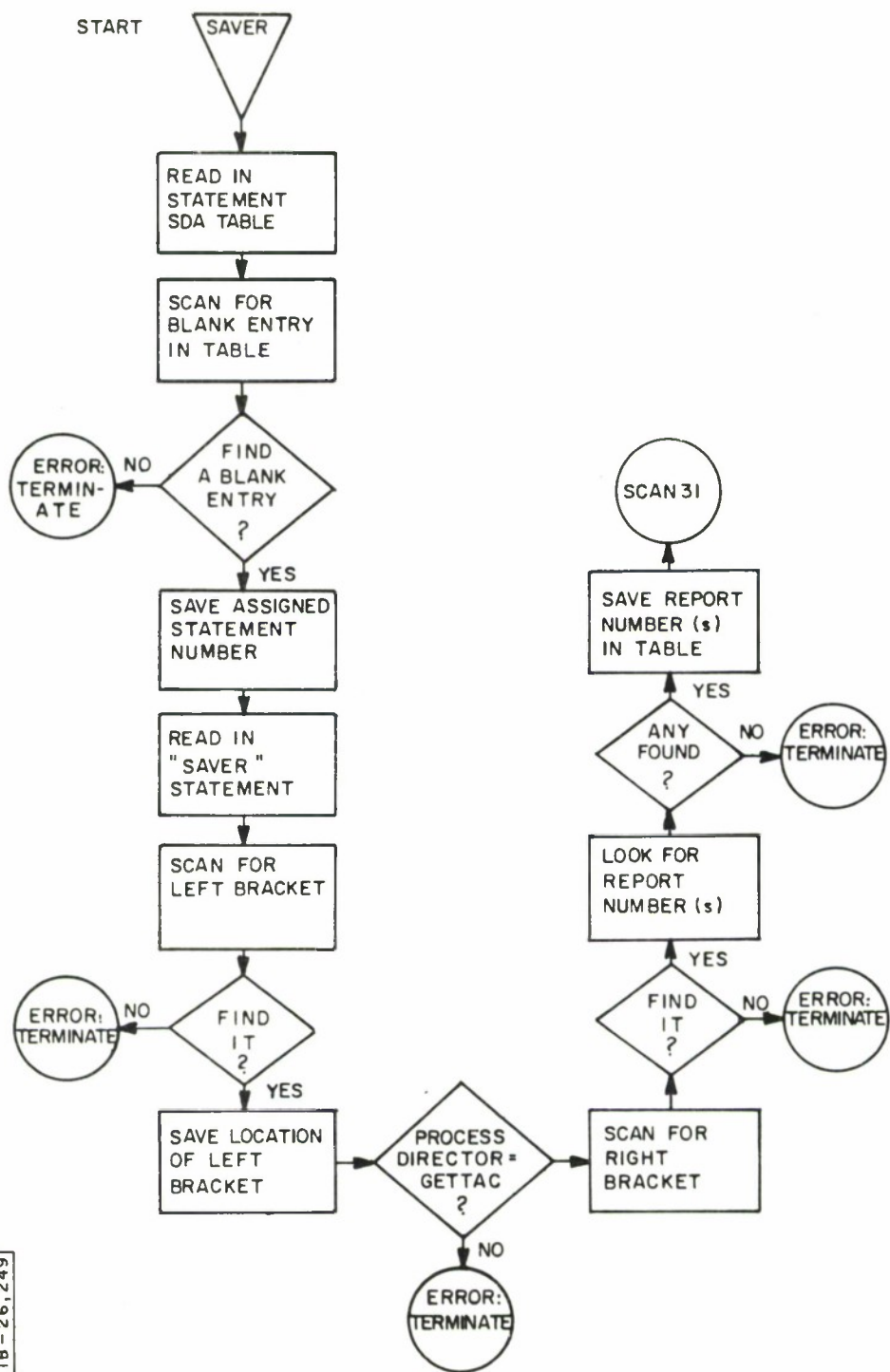
BB''' = output selector

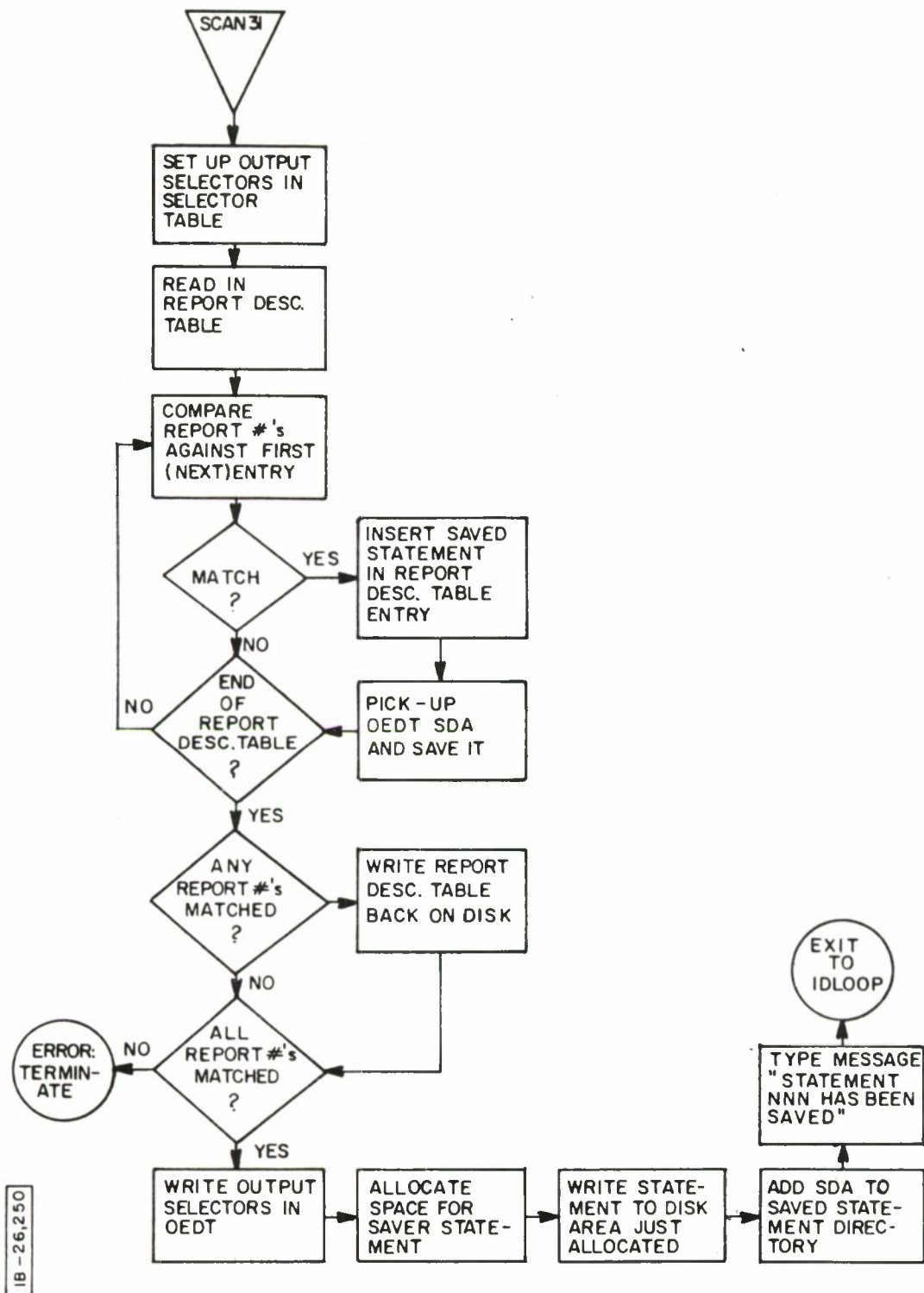
XXX = report numbers (maximum of 10)

Description

The saved statement SDA table is read from disk. The first blank entry is located and noted. The statement to be saved is accessed from the buffer SDA '0003'. The statement is scanned for correctness, such as appearance of a left and right bracket and report numbers. The report numbers are stored in a small table area. The output selectors are stored in a table area 50 characters to each entry. The report descriptor file is now accessed. Report numbers are now matched from the report descriptor file with those of the saved statement. If a match is found the saved statement number is inserted in the appropriate slot of the report descriptor file. The output entry descriptor table SDA and number of records referenced by the report descriptor is saved in a table area. The new report descriptor is written back on system disk. The process is continued until all report numbers from the saved statement are exhausted.

The output selector table is now written on disk at the SDA's indicated by the matching reports discovered. Disk space is allocated for the saved statement and the statement is saved at the SDA allocated. The actual SDA for the statement is inserted in the SDA statement buffer and rewritten on the system disk. The program will terminate with a printout of the statement number assigned to the 'SAVER' statement.





LOD

Purpose

LOD processes an Output Program, checking the output format statements syntactically, and condensing them into a convenient form for interpretation and execution by the output phase of REPGEN.

Calling Sequence

Enter LOD on the console typewriter. Return will be to IDLOOP.

Description

The Output Descriptor Card is read and Report Descriptor Table is called from disk. A search is instituted for the report indicated on the Output Descriptor Card.

When the report is found, the SOT Table for this report is called in from disk and updated with the current information from the Output Descriptor Card. An SDA is now allocated for the condensed output format statements.

The SOT Table is now put back on disk with the present run recorded as in error status. The condensed output statement SDA is put into the appropriate DSKACC call.

The LOD statement cards are read in next and are condensed by LOD. Each numbered statement is allotted a complete 245 character record within the assigned SDA. When an attribute name is encountered during processing of the statement cards, an Output Entry Descriptor Table (OEDT -- an expandable 2800 character record) for this report is brought into core from disk. The attribute name is replaced by a six digit value from the OEDT. This value specifies the attribute value length and its relative position in the intermediate tape created by the first phase of REPGEN.

The OEDT usually will remain in core throughout the processing of the present statement cards except when an additional segment of the OEDT has to be called in from disk. On the next reference to the OEDT the first segment of the table will be brought back into core.

When the final statement has been processed, the report descriptor table will be written back onto disk, after being submitted to a minor update. This update involves putting an indicator for output media into the table.

A second wind-up operation puts the updated SOT Table onto disk. A successful set of statements get an "A" or "C" in position 0 of the table. Otherwise an "E" is left in the table.

Phrase Types and Condensed Form

Type 1

DO XXX	1DXXX
REPEAT XXX	1RXXX
GO TO XXX	1GXXX
CONTINUE XXX	1CXXX

Type 2

SET OVERFLOW AFTER NN	2SONN
SET SWITCH N	2SSN
SET COUNTER NN LENGTH=PP	2SCNNPP
SET REMAINDER LENGTH=NN	2SC00NN
CLEAR SWITCH N	2CSN
CLEAR SWITCHES	2CS
CLEAR COUNTER NN	2CCNN
CLEAR COUNTERS	2CC
CLEAR	2

Type 3

SKIP TO A NEW PAGE	3E1
SKIP NN	3CNN
STACKER N	3SN
WRITE TAPE MARK	3T

Type 4

END LINE	40
END CARD	40
END HEADER XXX	4HXXX
END TRAILER XXX	4TXXX
END ENTRY	40
NEXT ENTRY	4
END	1

Type 5

IF OVERFLOW	50
IF FINISHED	5F

Type 5 (continued)

IF SWITCH X

$$\begin{array}{c} \text{N} \\ \text{IF } b \text{ (....)} = \text{N} \text{ (....)} / \\ \text{D} \quad \quad \quad \text{U} \text{ D} \quad \quad \quad \text{D} \end{array}$$

5SX

$$\begin{array}{c} \text{N} \quad \quad \quad \text{U} \\ 5 \text{ b} \text{} = \text{S} \text{ N} \text{} \\ \text{D} \quad \quad \quad \text{T} \text{ D} \end{array}$$

Type 6

$$\begin{array}{c} \text{N} \\ \text{COL XXX} \text{ L} = \text{D} \text{ (....)} \\ \text{R} \quad \quad \quad \text{C} \\ \quad \quad \quad \text{b} \end{array}$$

$$\begin{array}{c} \text{N} \\ 6\text{XXX} \text{ L} = \text{D} \text{} \\ \text{R} \quad \quad \quad \text{C} \\ \quad \quad \quad \text{b} \end{array}$$

$$\text{COL XXX} = \text{N} \text{ (....)} \\ \text{S}$$

$$6\text{XXXR} \text{ N} \text{} \\ \text{S}$$

$$\text{COL XXX} = \text{D} \text{ (....)} \\ \text{C} \\ \text{b}$$

$$6\text{XXXL} \text{ D} \text{} \\ \text{C} \\ \text{b}$$

$$\text{COL XXX} \text{ L} = \text{N} \text{ (N(....) } \frac{+}{-} \text{ N (....))} \\ \text{R} \quad \text{S} \quad \text{b} \quad \text{b} \quad \text{x}$$

$$6\text{XXX} \text{ L} \text{ N} \text{} \text{ A} \text{ N} \text{} \\ \text{R} \quad \text{S} \quad \text{b} \quad \text{S} \quad \text{b} \quad \text{b}$$

$$\text{COL XXX} = \text{N} \text{ (N(....) } \frac{+}{-} \text{ N (....))} \\ \text{S} \quad \text{b} \quad \text{b} \quad \text{b} \quad \text{x}$$

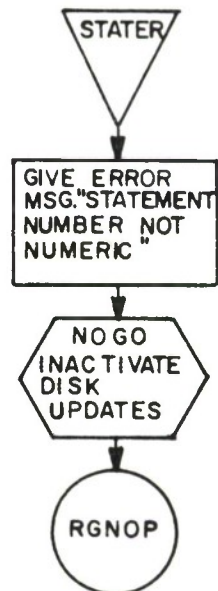
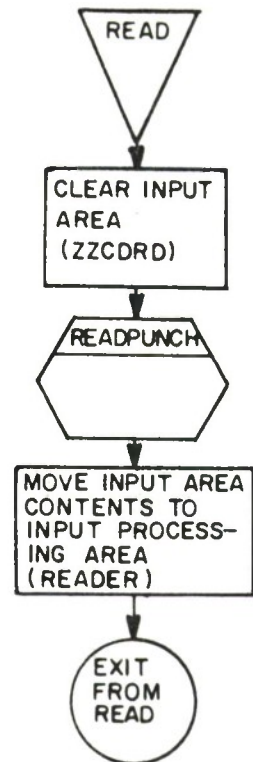
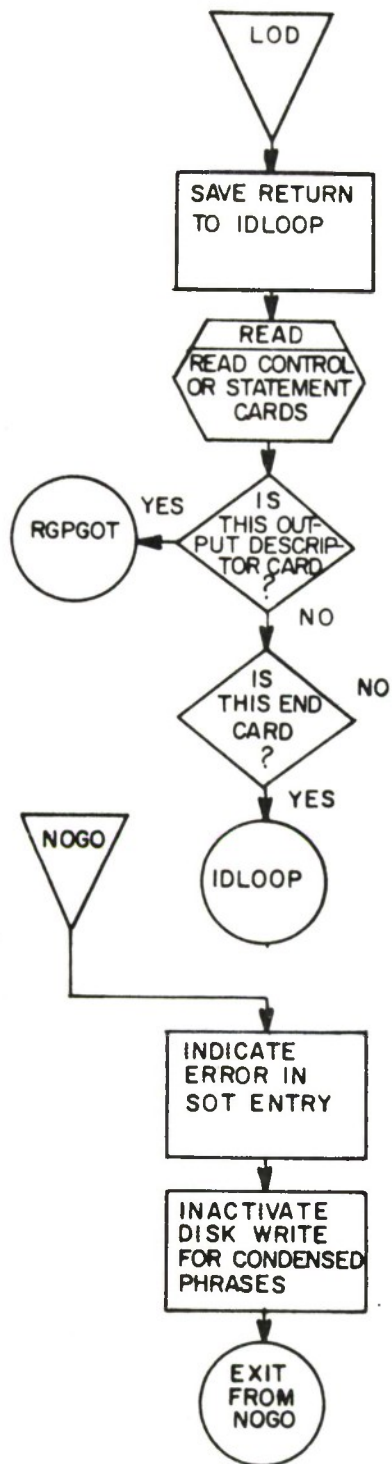
$$6\text{XXXR} \text{ N} \text{ N} \text{} \text{ A} \text{ S} \text{ N} \text{} \\ \text{S} \quad \text{b} \quad \text{S} \quad \text{b} \quad \text{b}$$

$$\text{COUNTER XX} = \text{N} \text{ (....)} \\ \text{D} \\ \text{b}$$

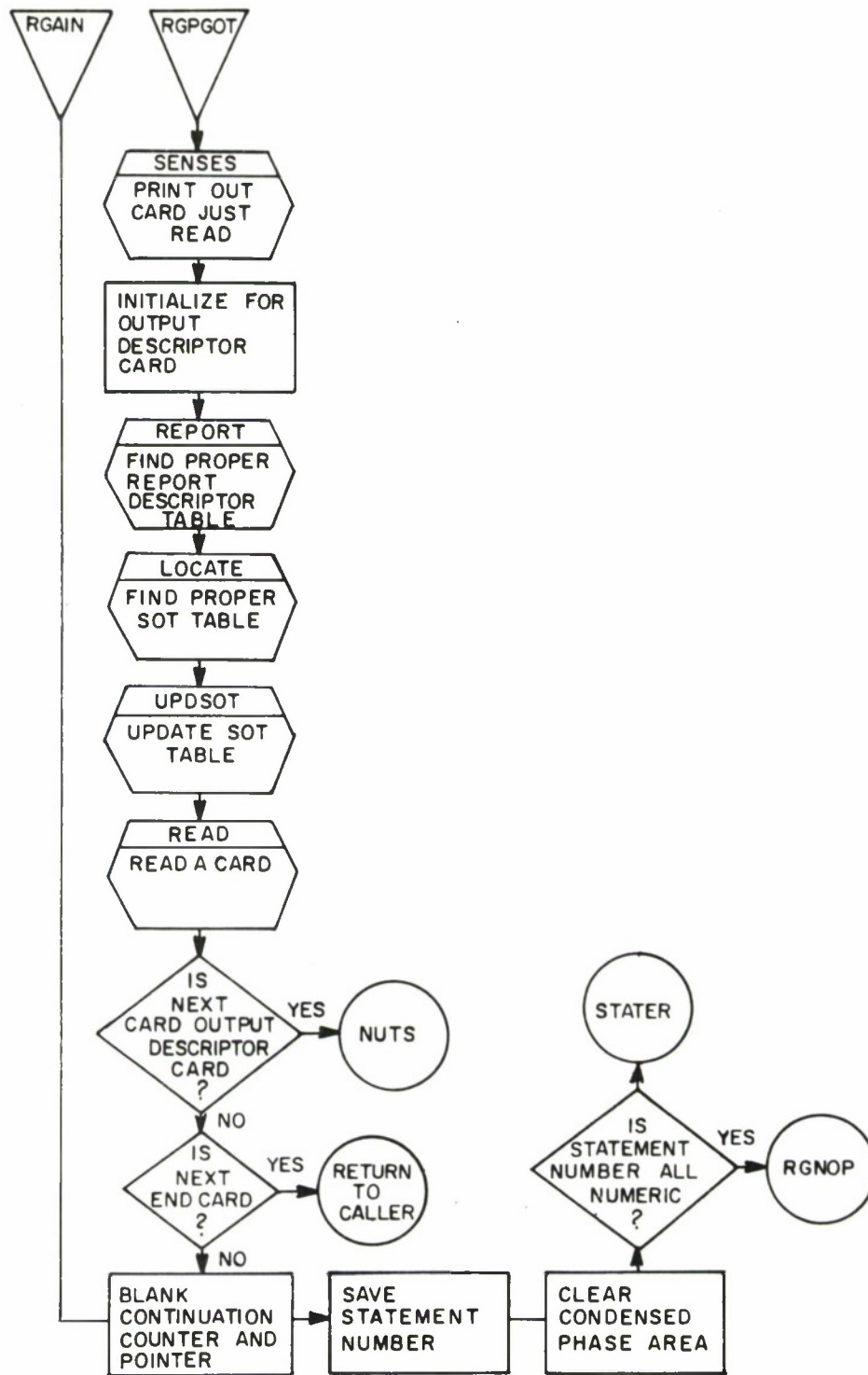
$$6\text{CXX} \text{ N} \text{ (....)} \\ \text{D} \\ \text{b}$$

$$\text{COUNTER XX} = \text{N} \text{ (N(....) } \frac{+}{-} \text{ N (....))} \\ \text{S} \quad \text{b} \quad \text{b} \quad \text{b} \quad \text{x}$$

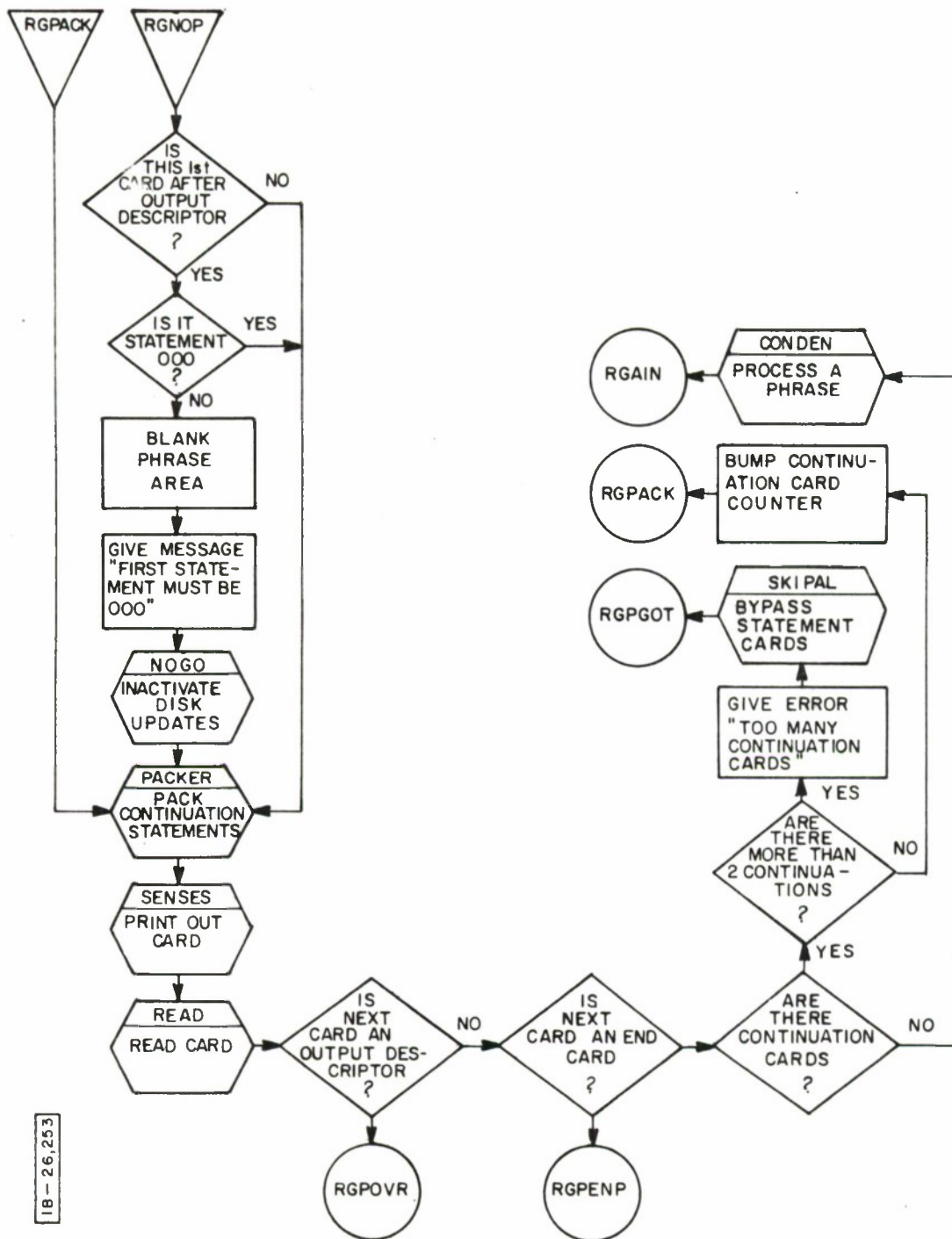
$$6\text{CXX} \text{ N} \text{ N} \text{} \text{ A} \text{ S} \text{ N} \text{} \\ \text{S} \quad \text{b} \quad \text{S} \quad \text{b} \quad \text{b}$$



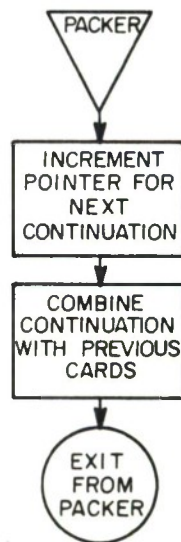
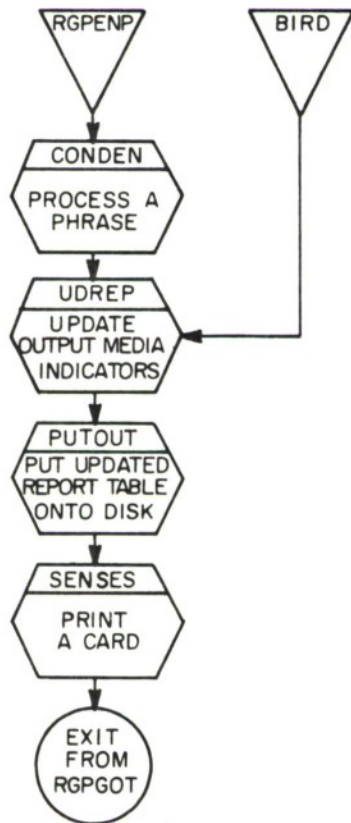
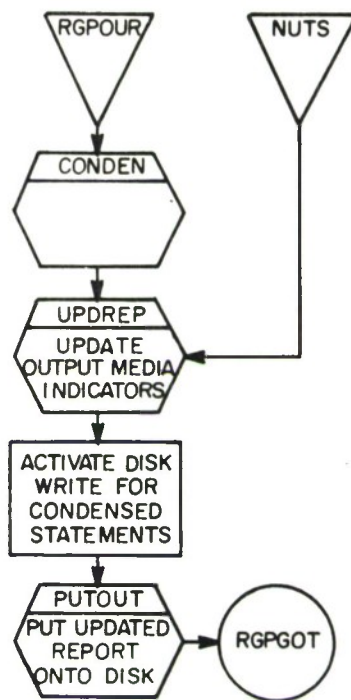
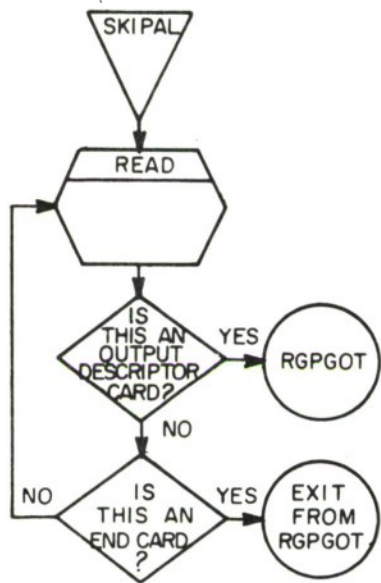
IB-26,251



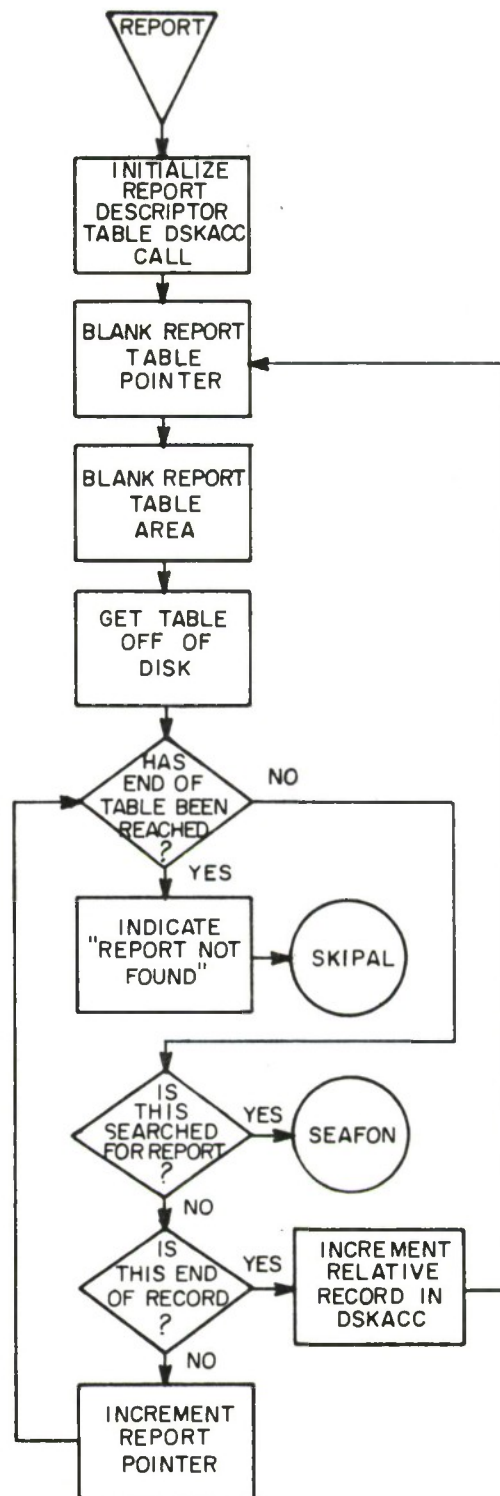
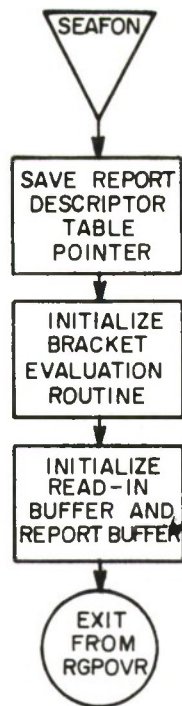
IB - 26,252



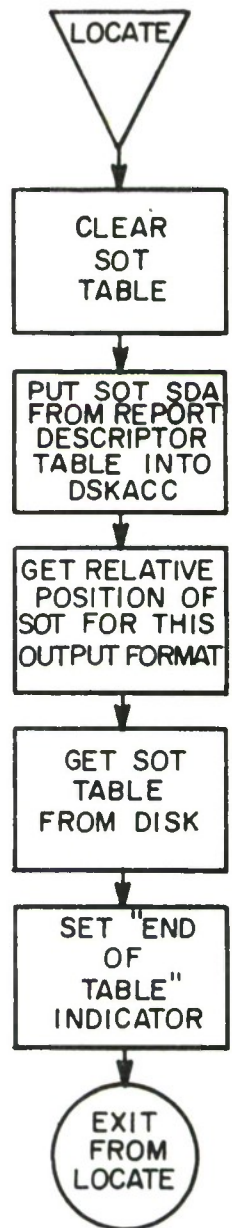
IB - 26,253



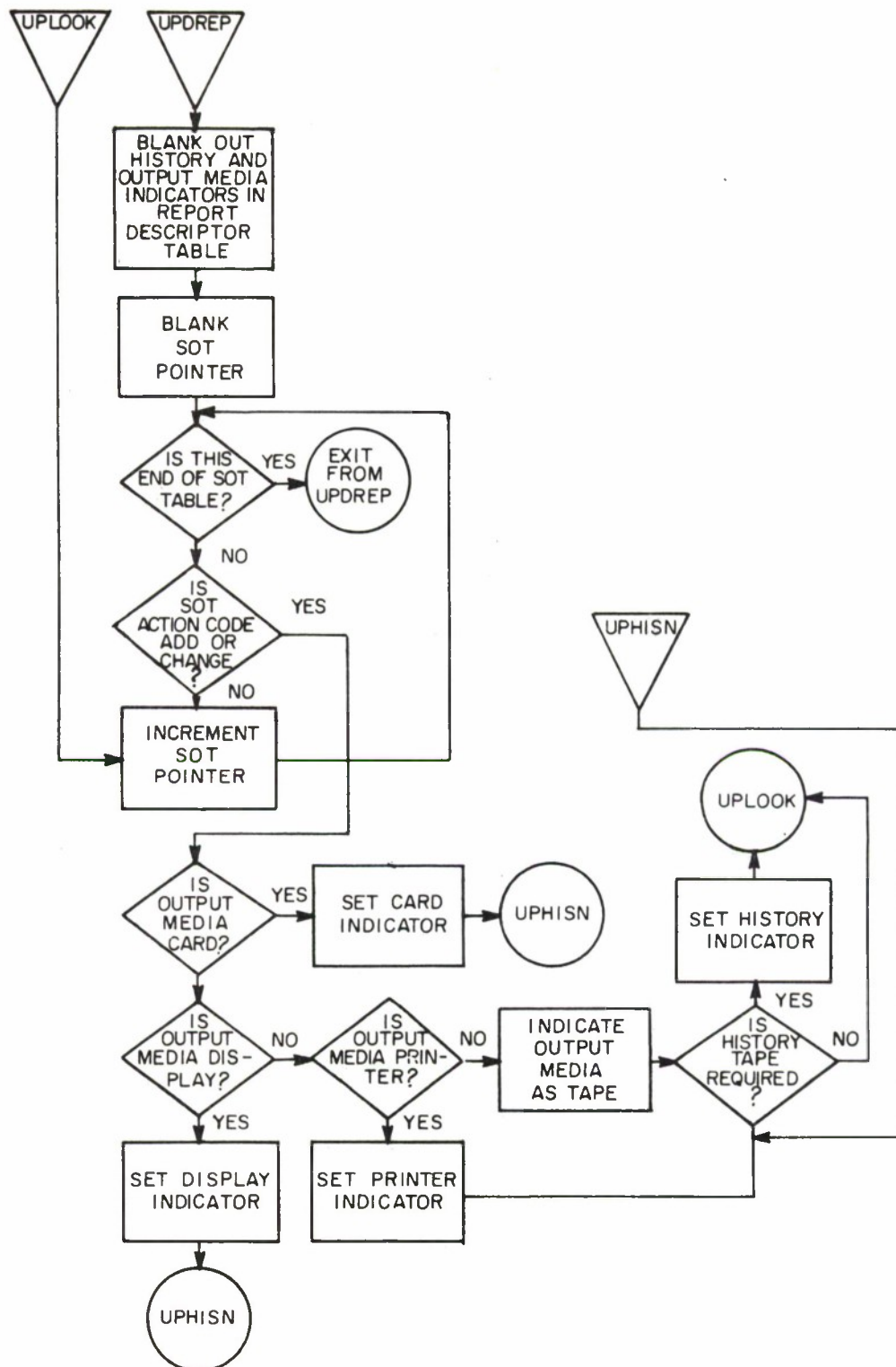
IB - 26, 198



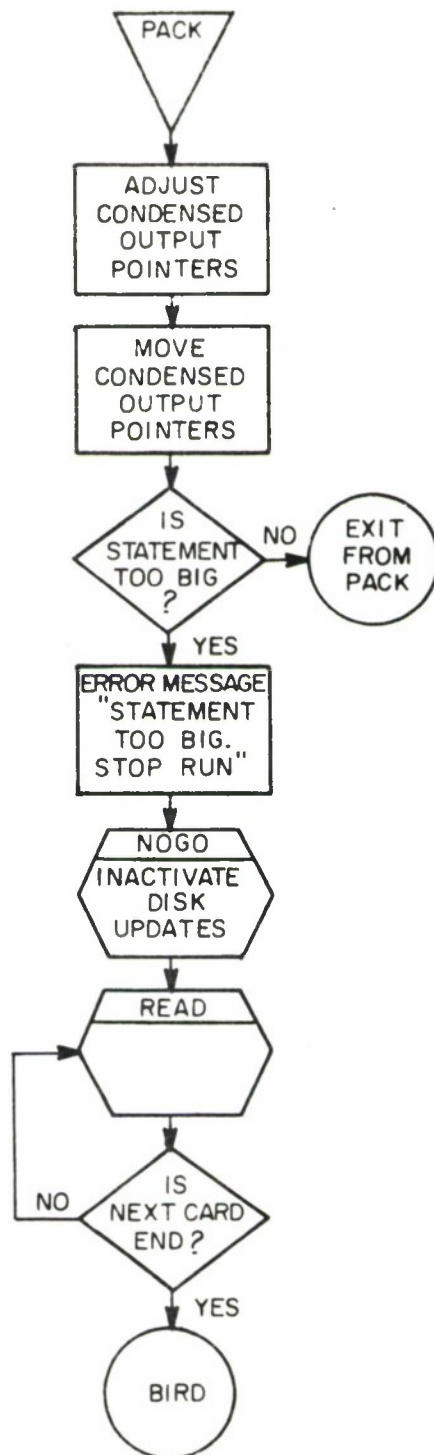
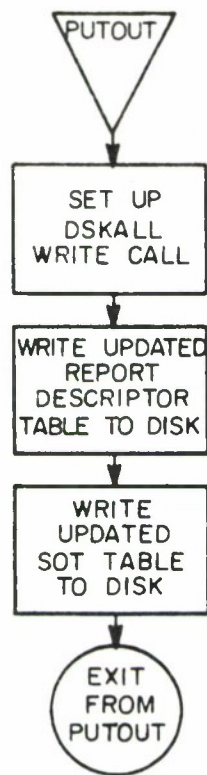
1B-26, 197



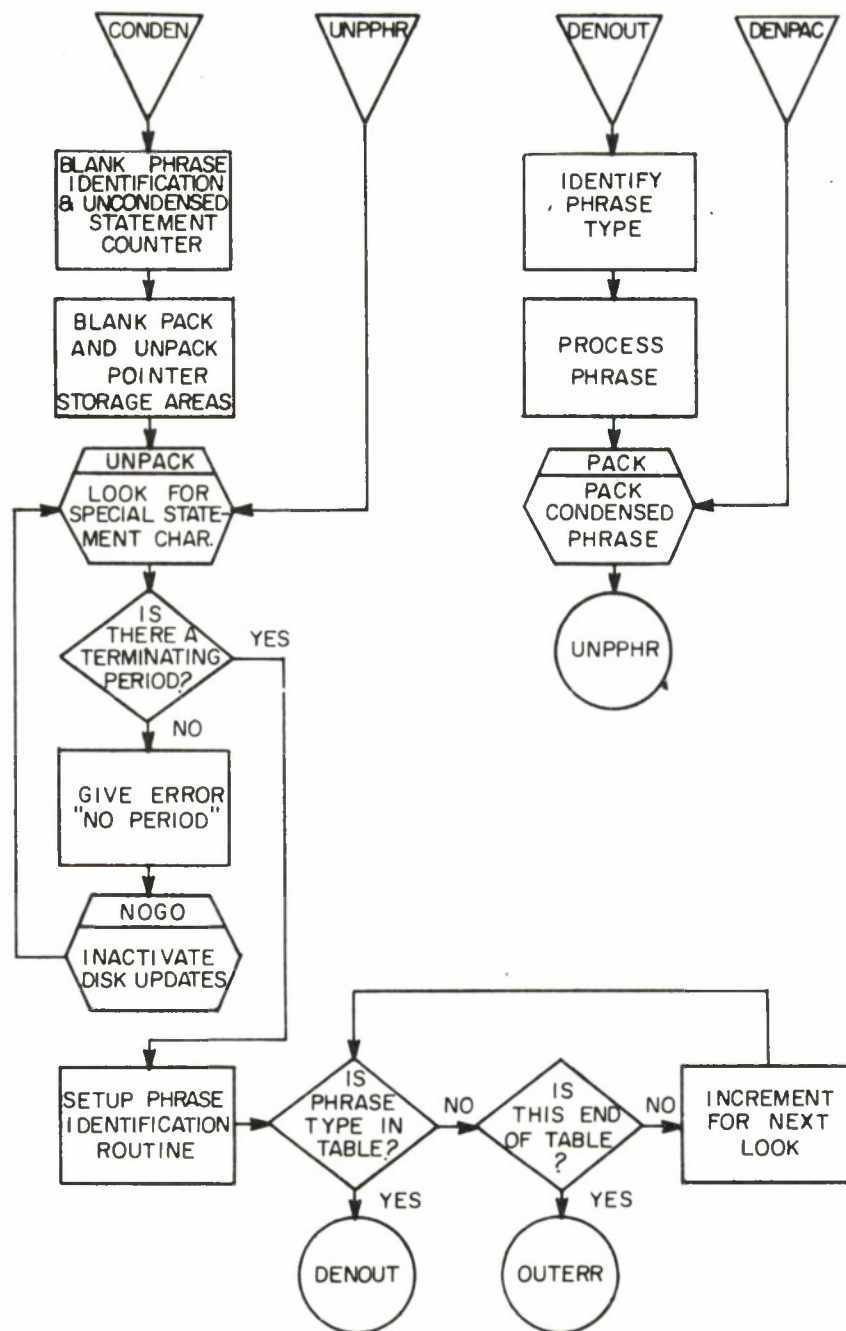
IB-26,193



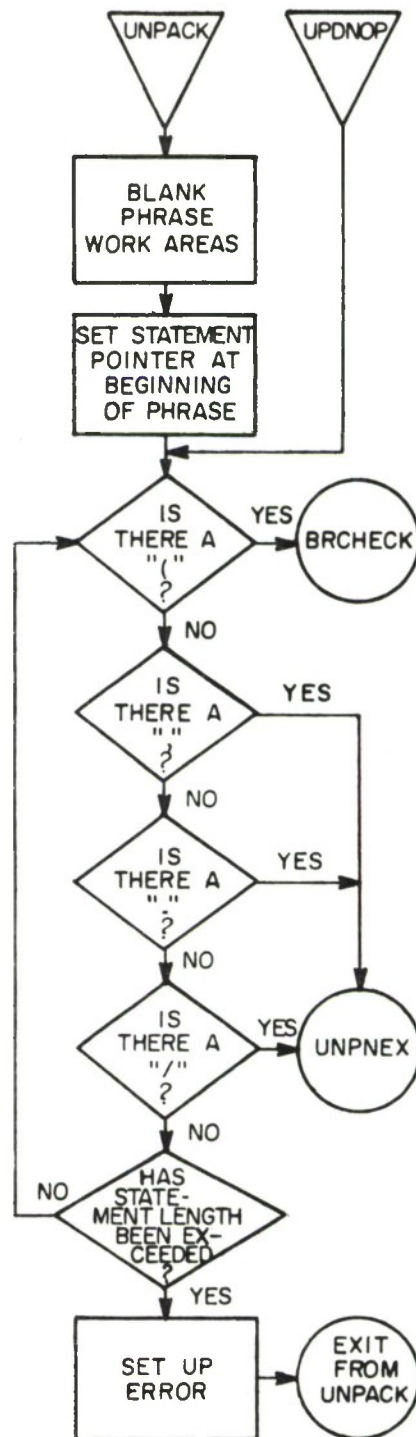
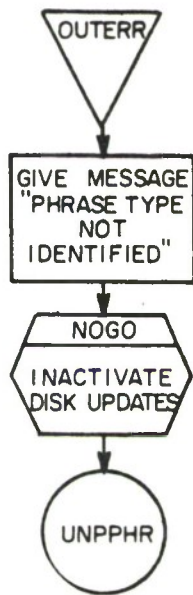
IB - 26, 194



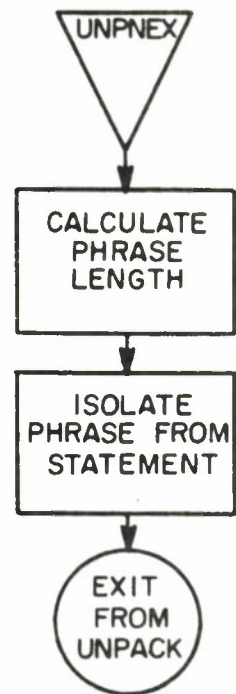
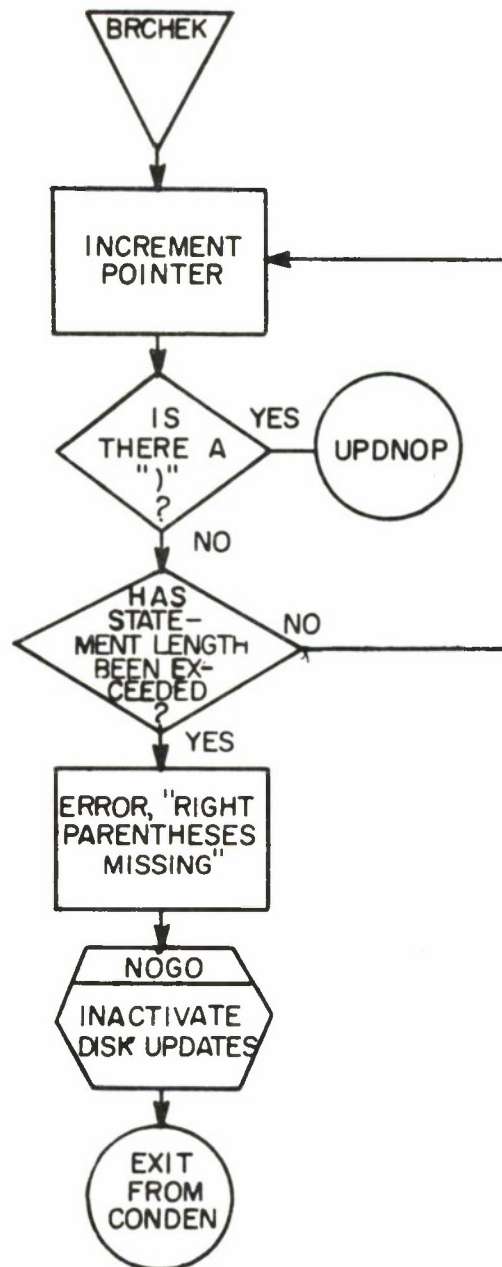
IB - 26,195



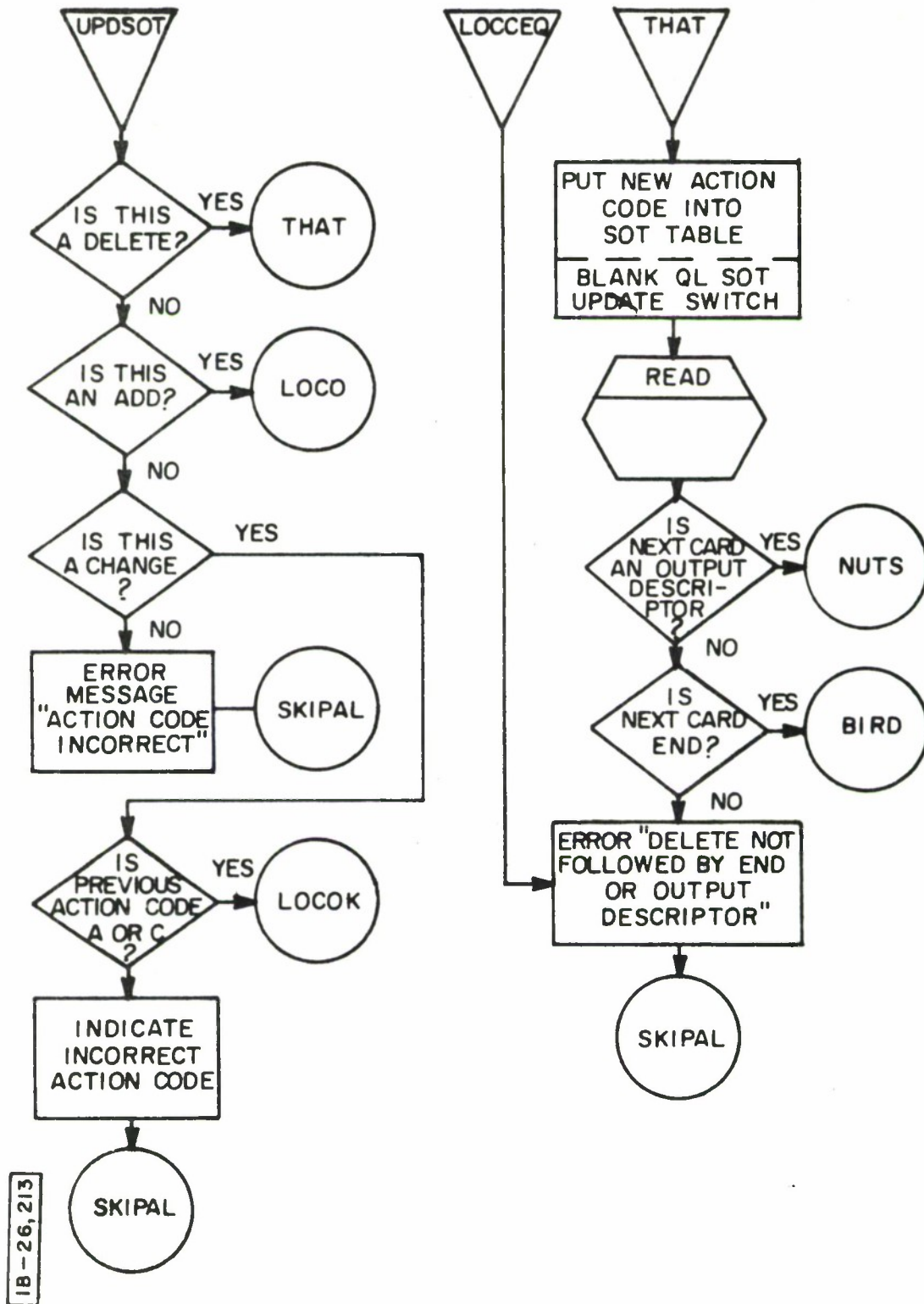
IB-26, 196

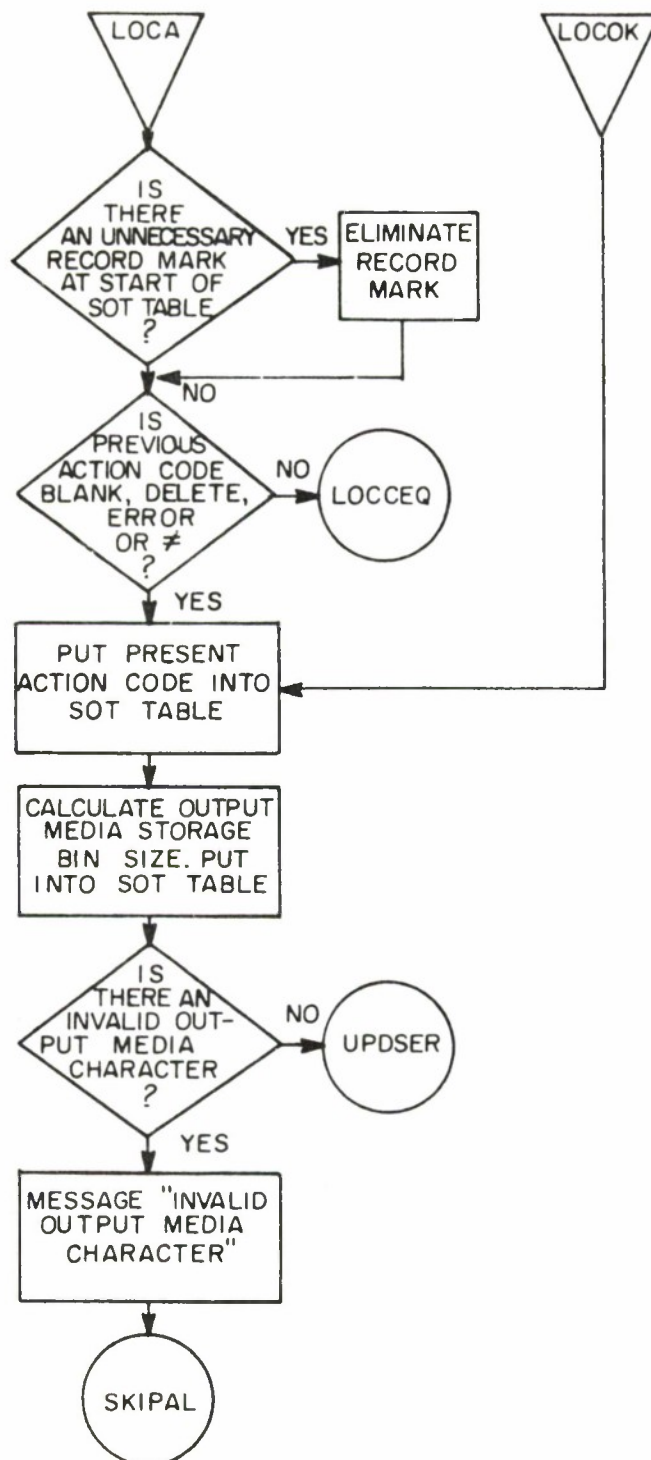


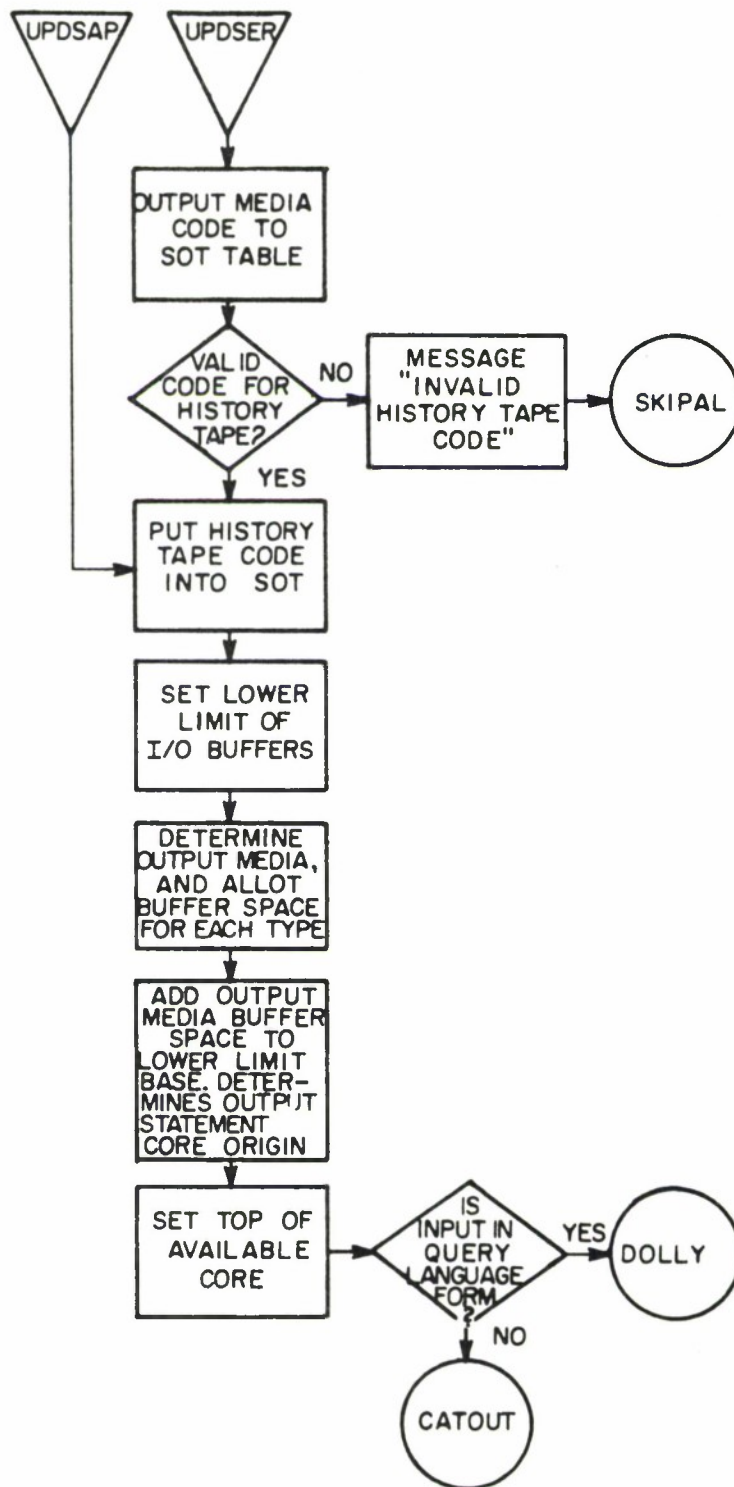
18-26,190



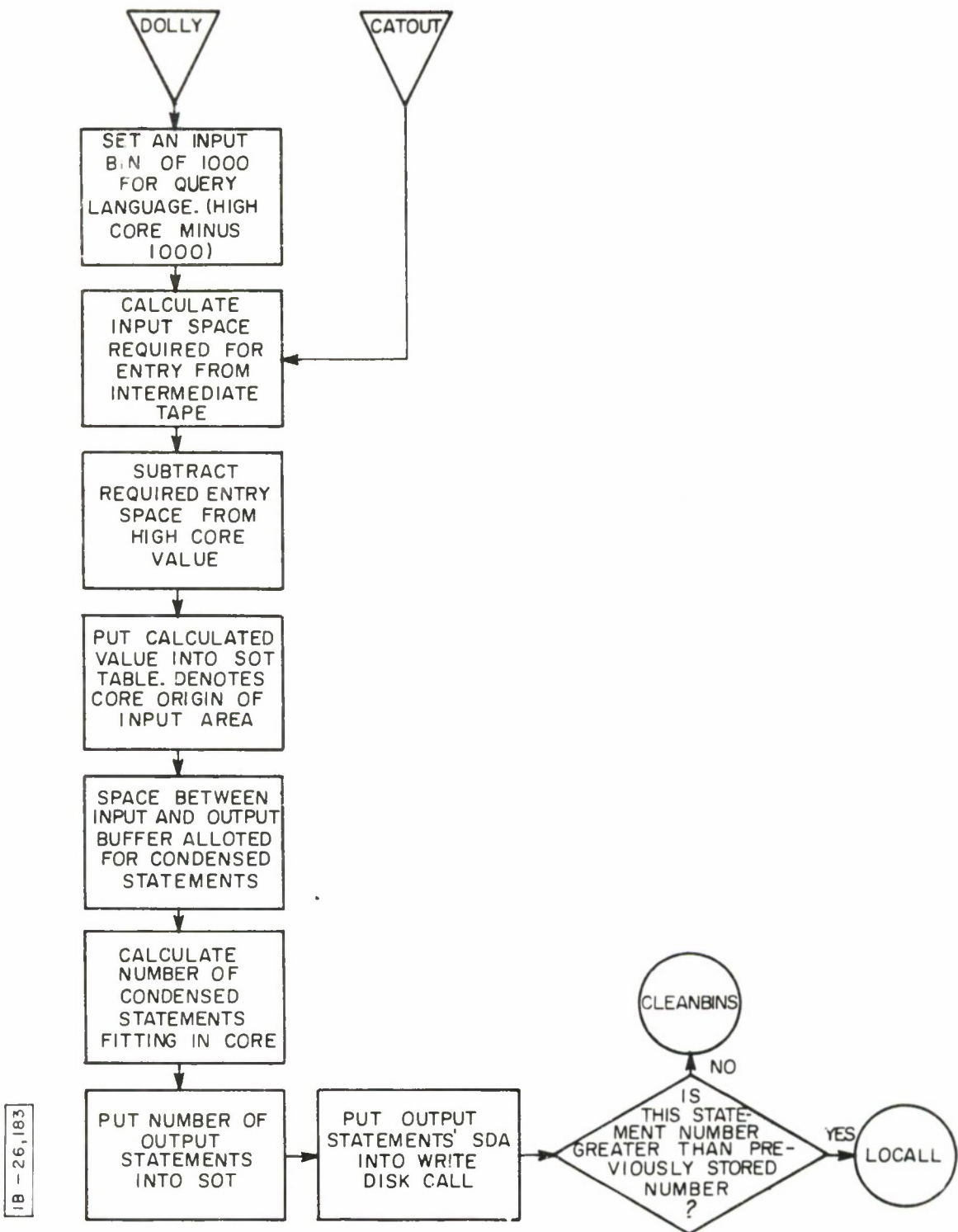
18-26,191

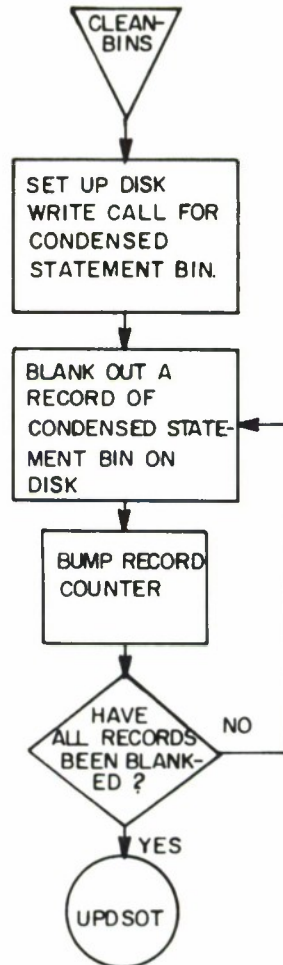
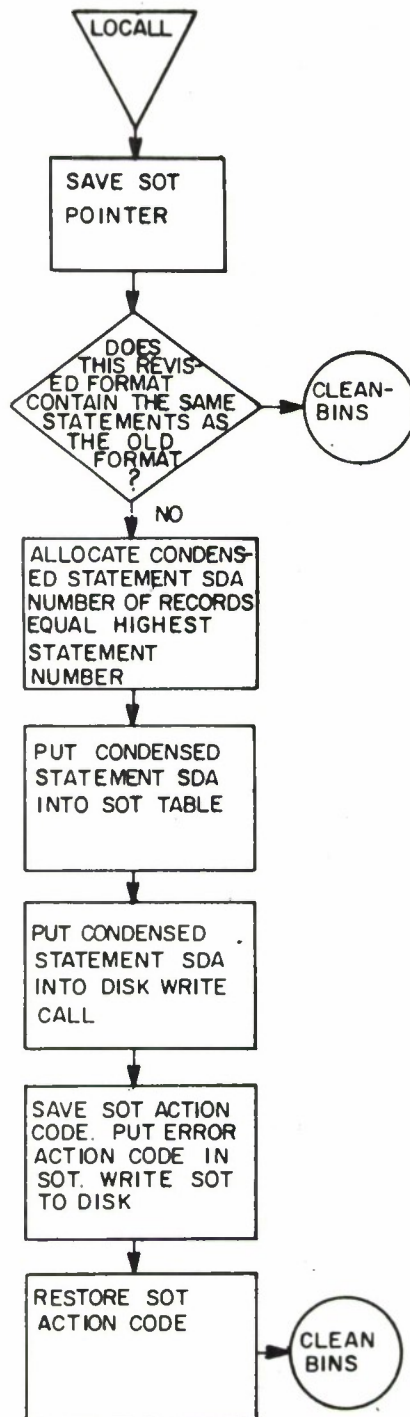




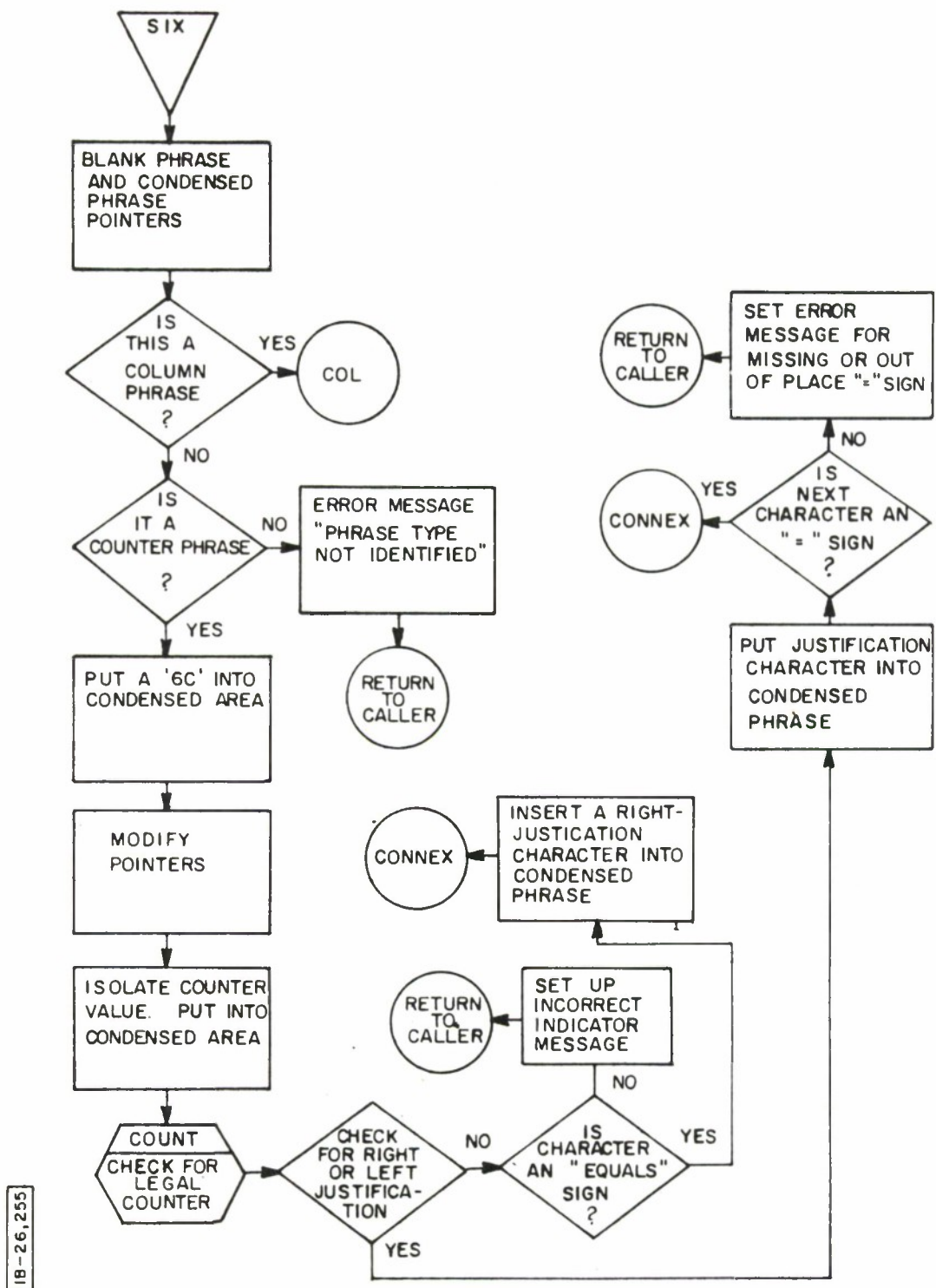


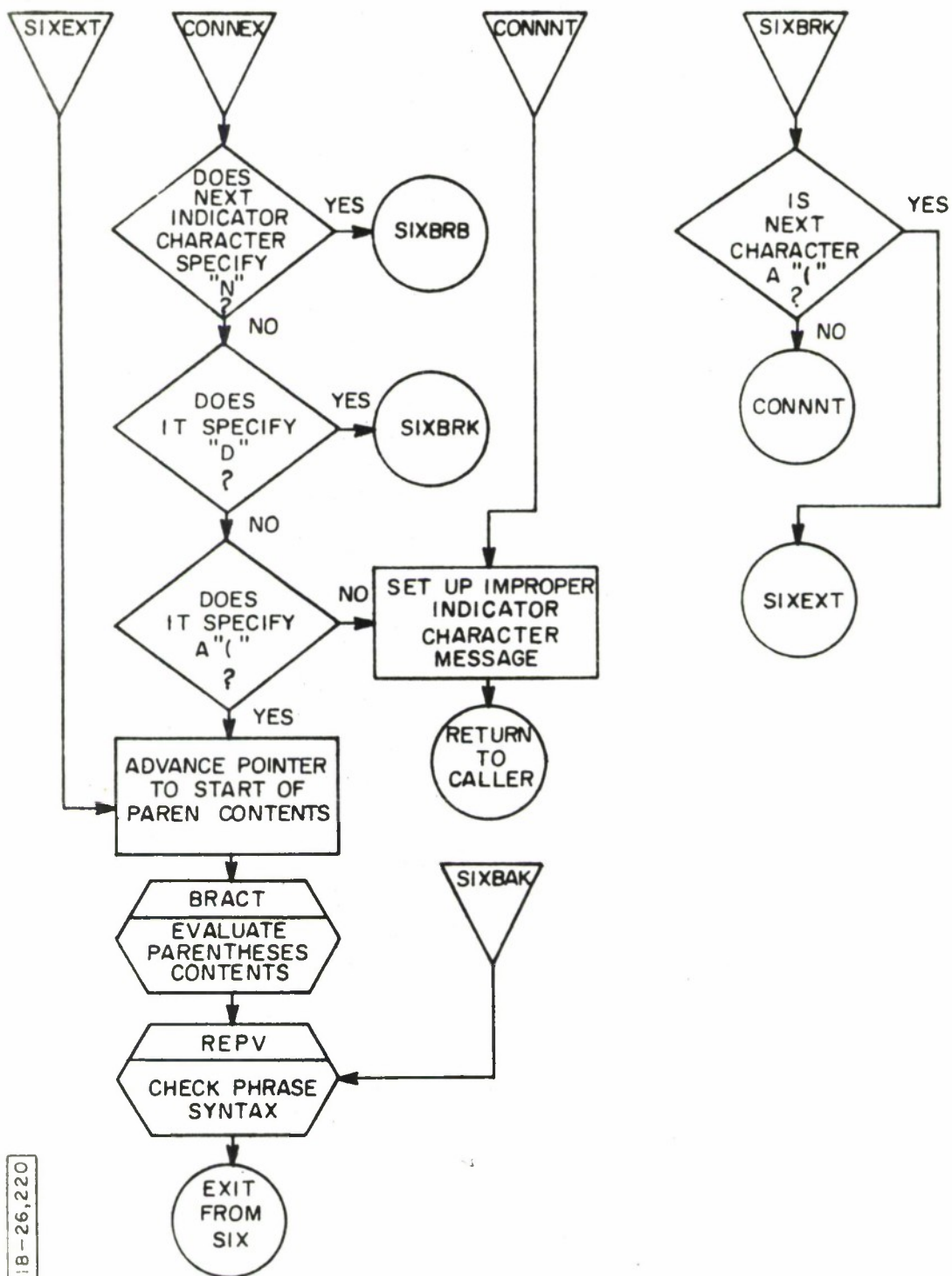
18-26,182



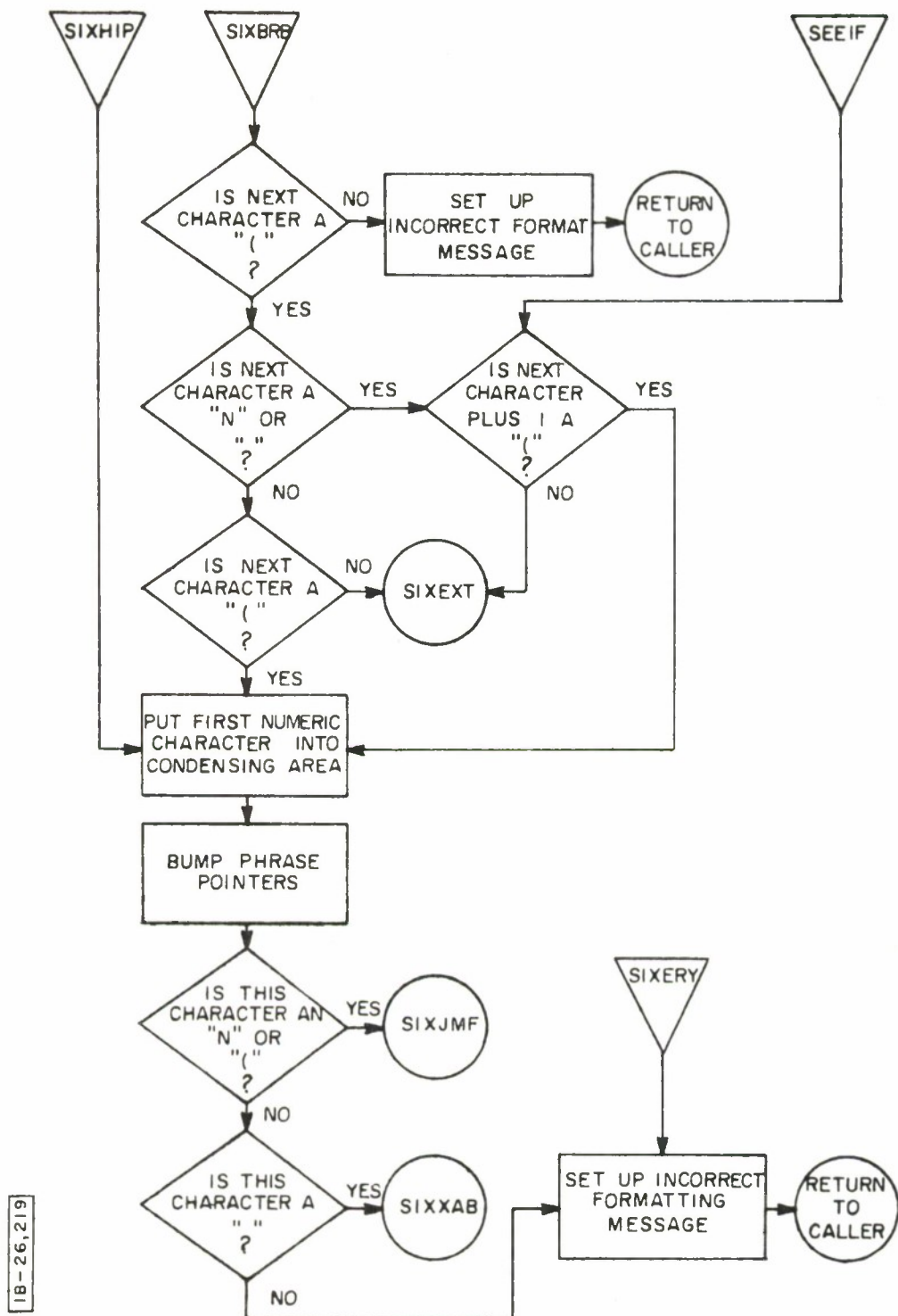


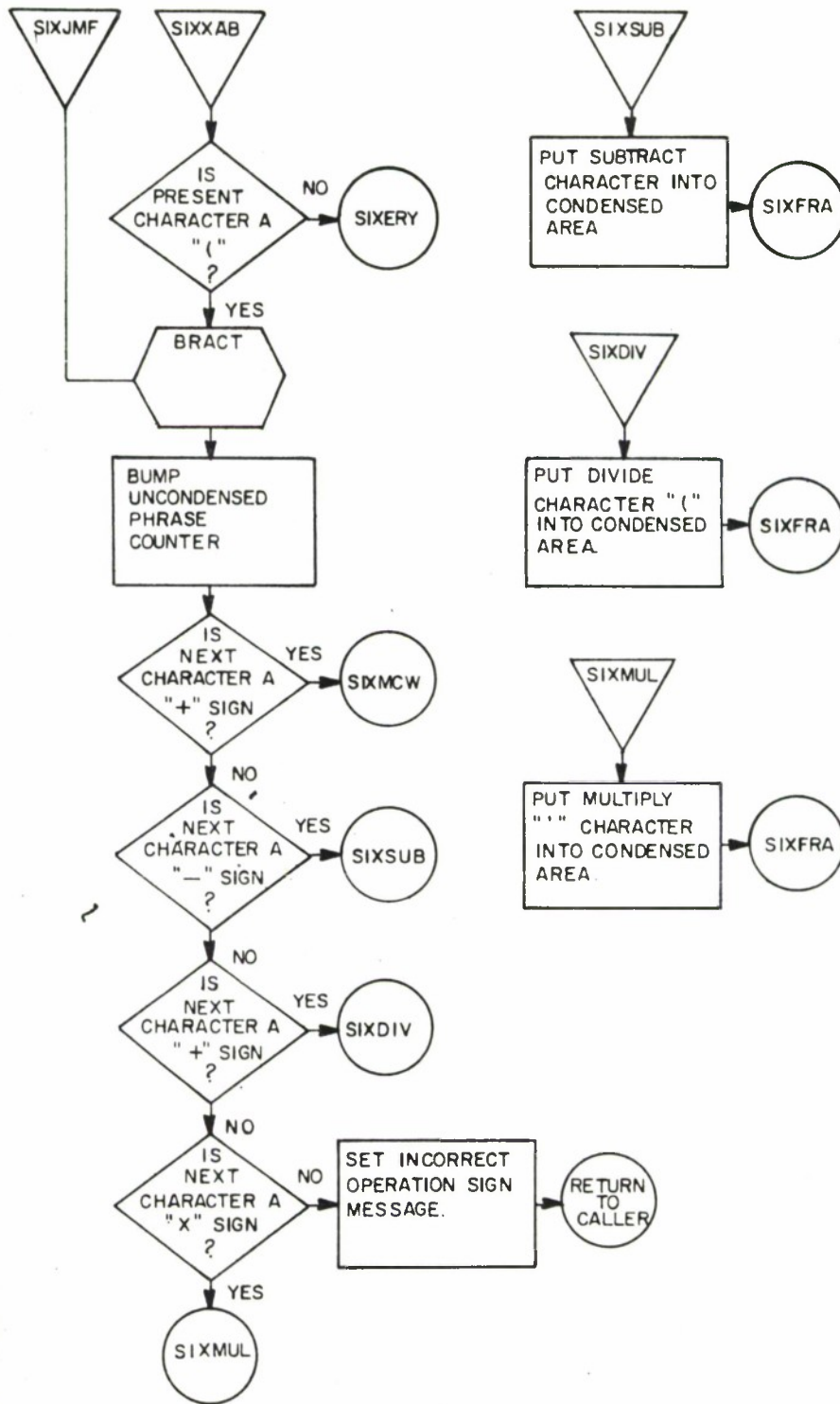
IB - 26,254



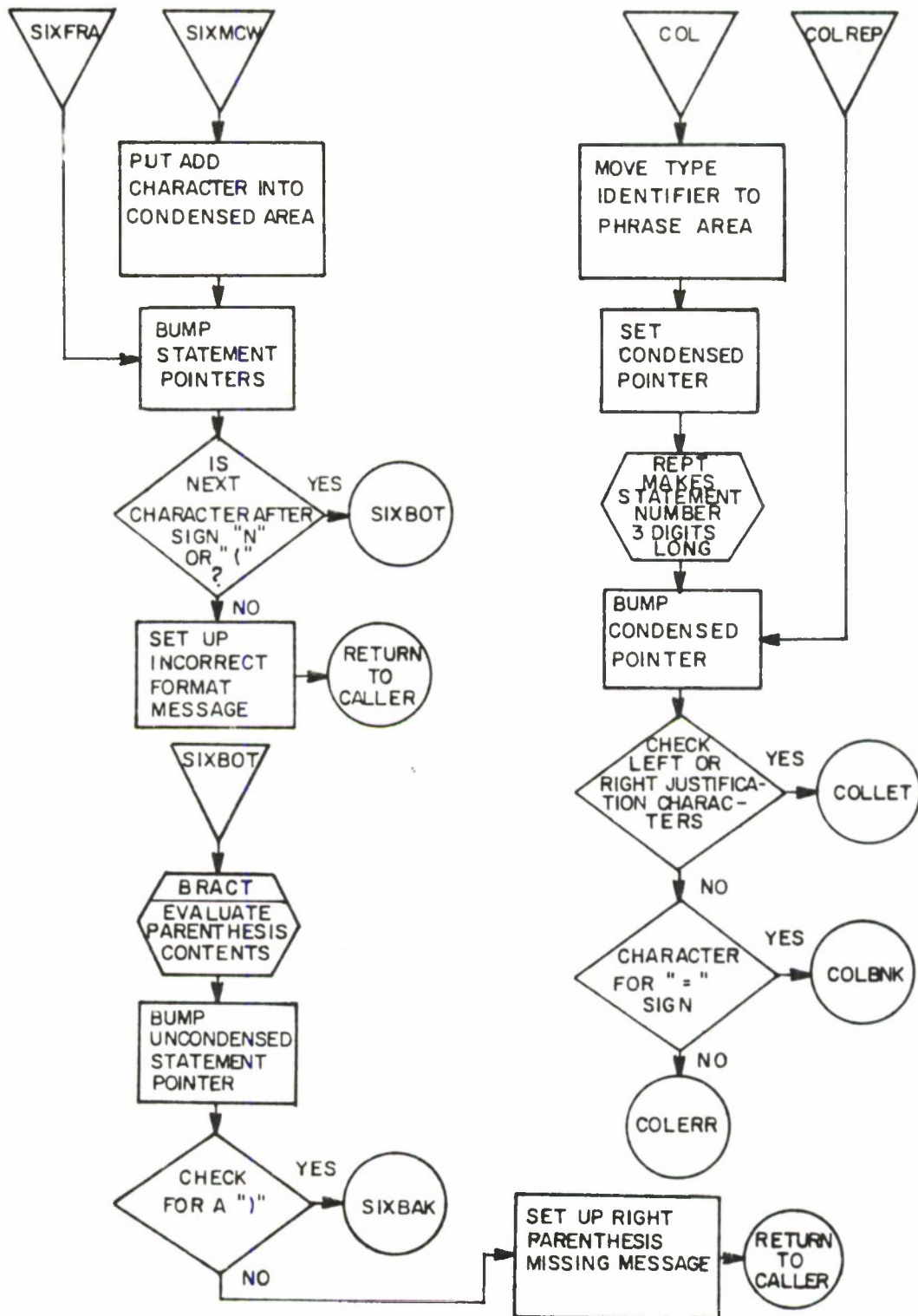


18-26,220

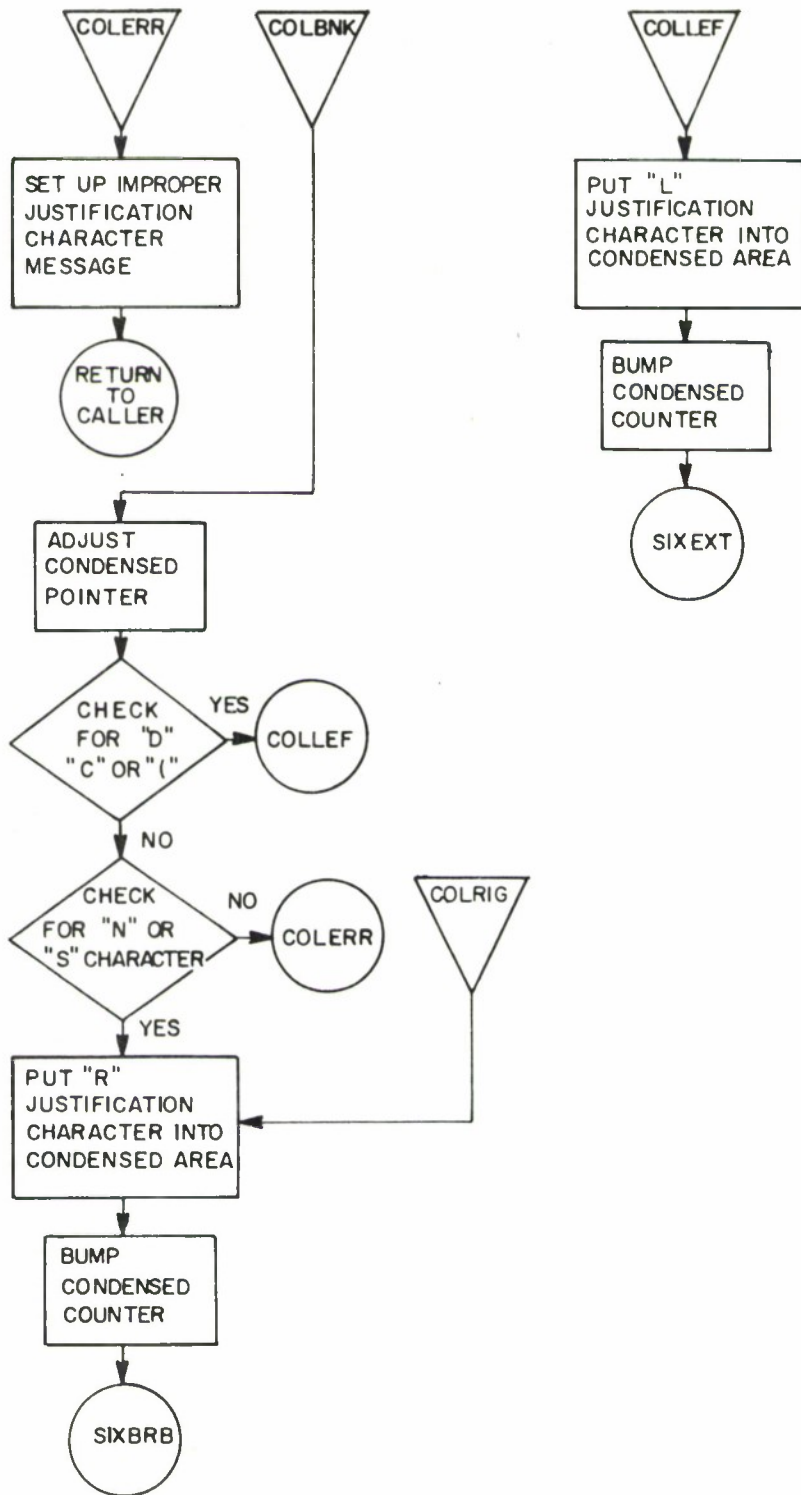




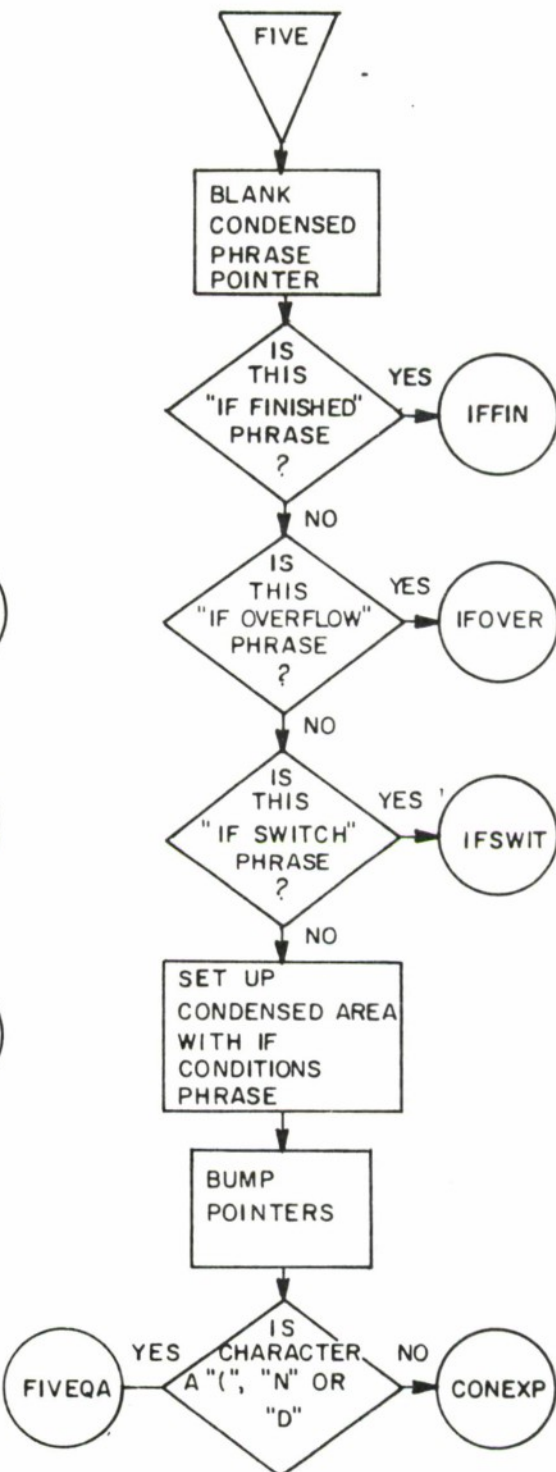
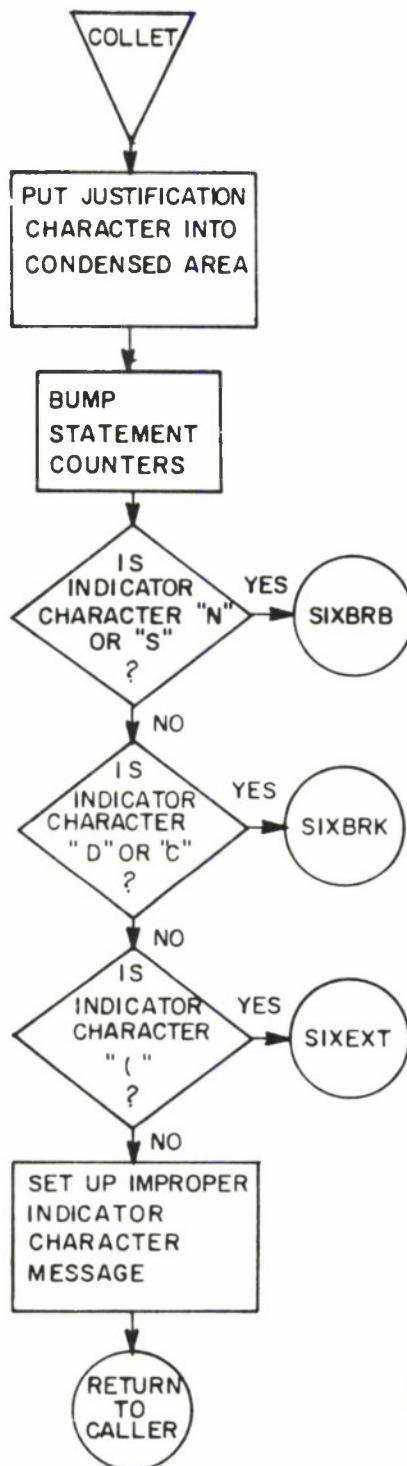
18-26,256



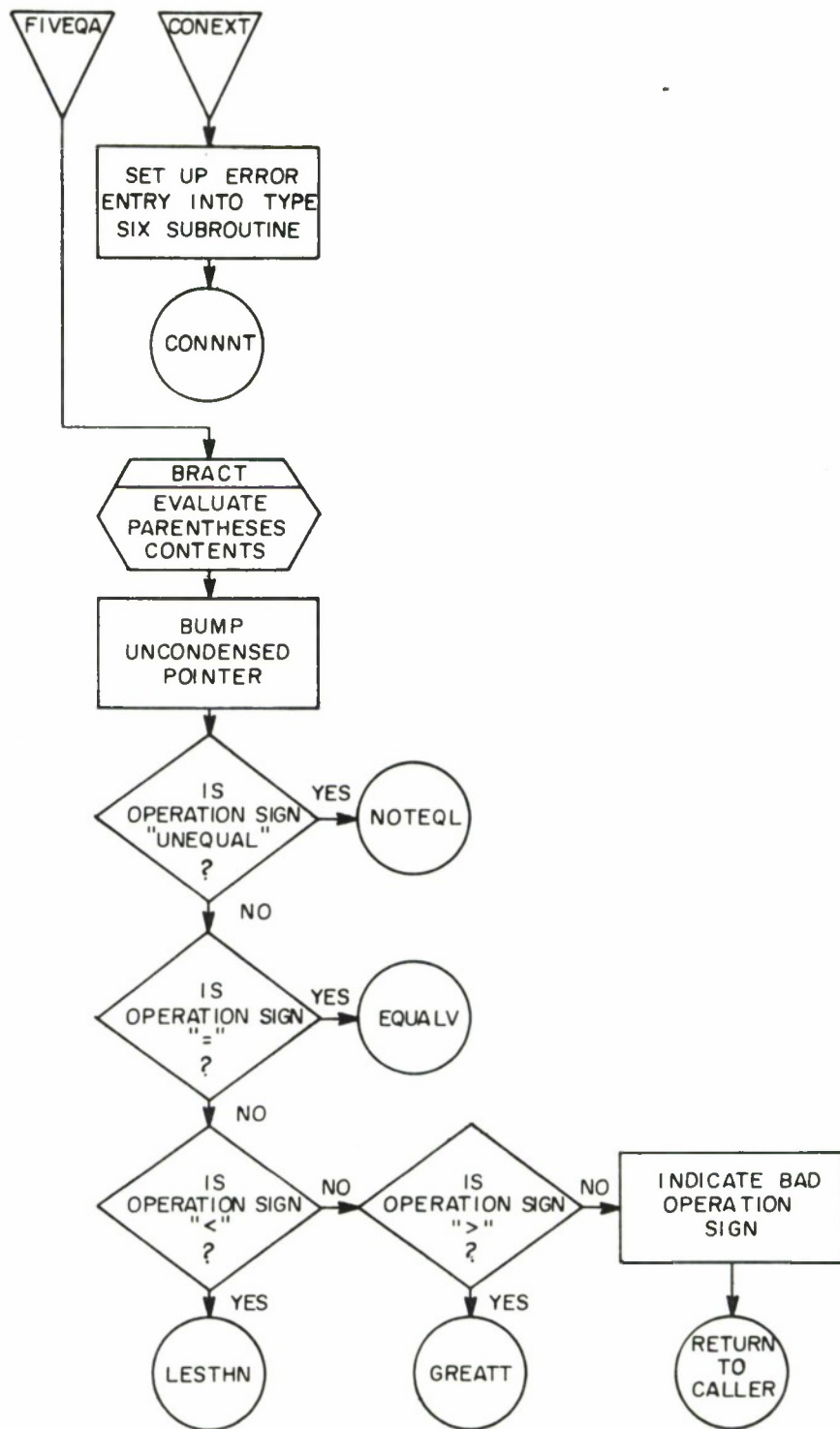
IB - 26,257



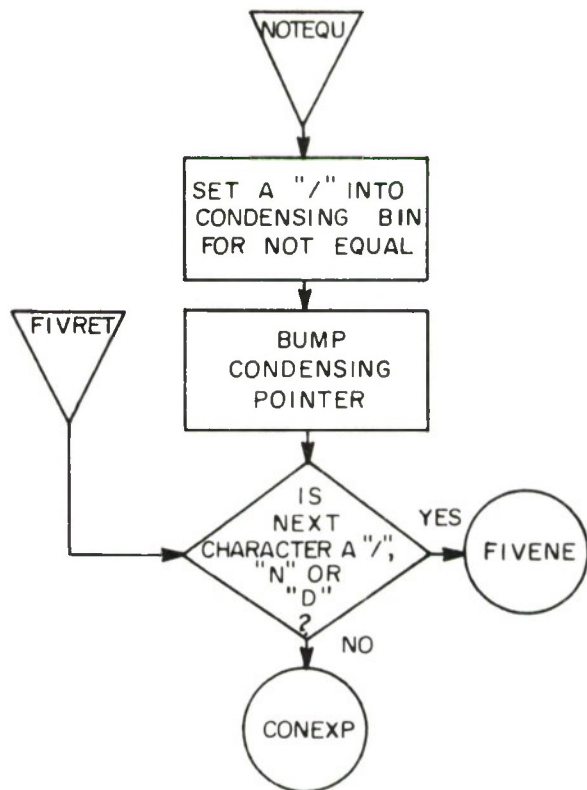
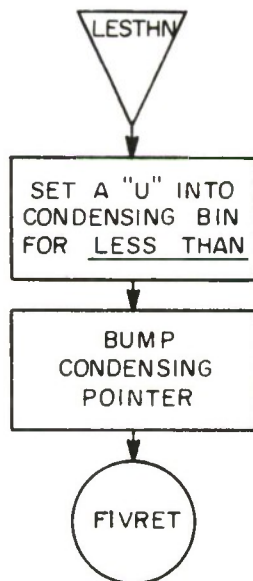
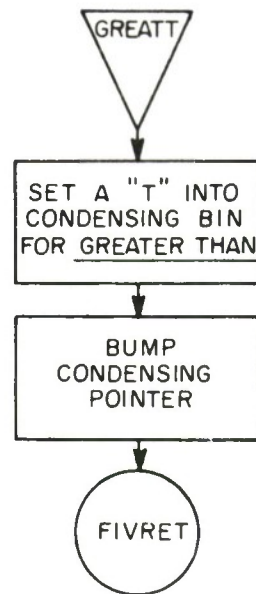
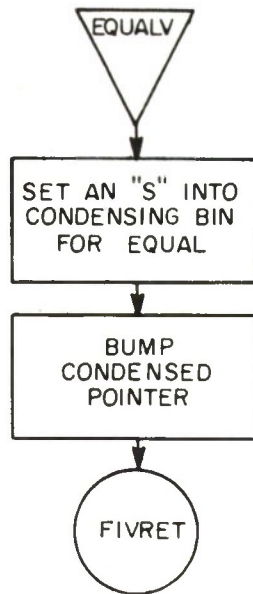
1B-26,258



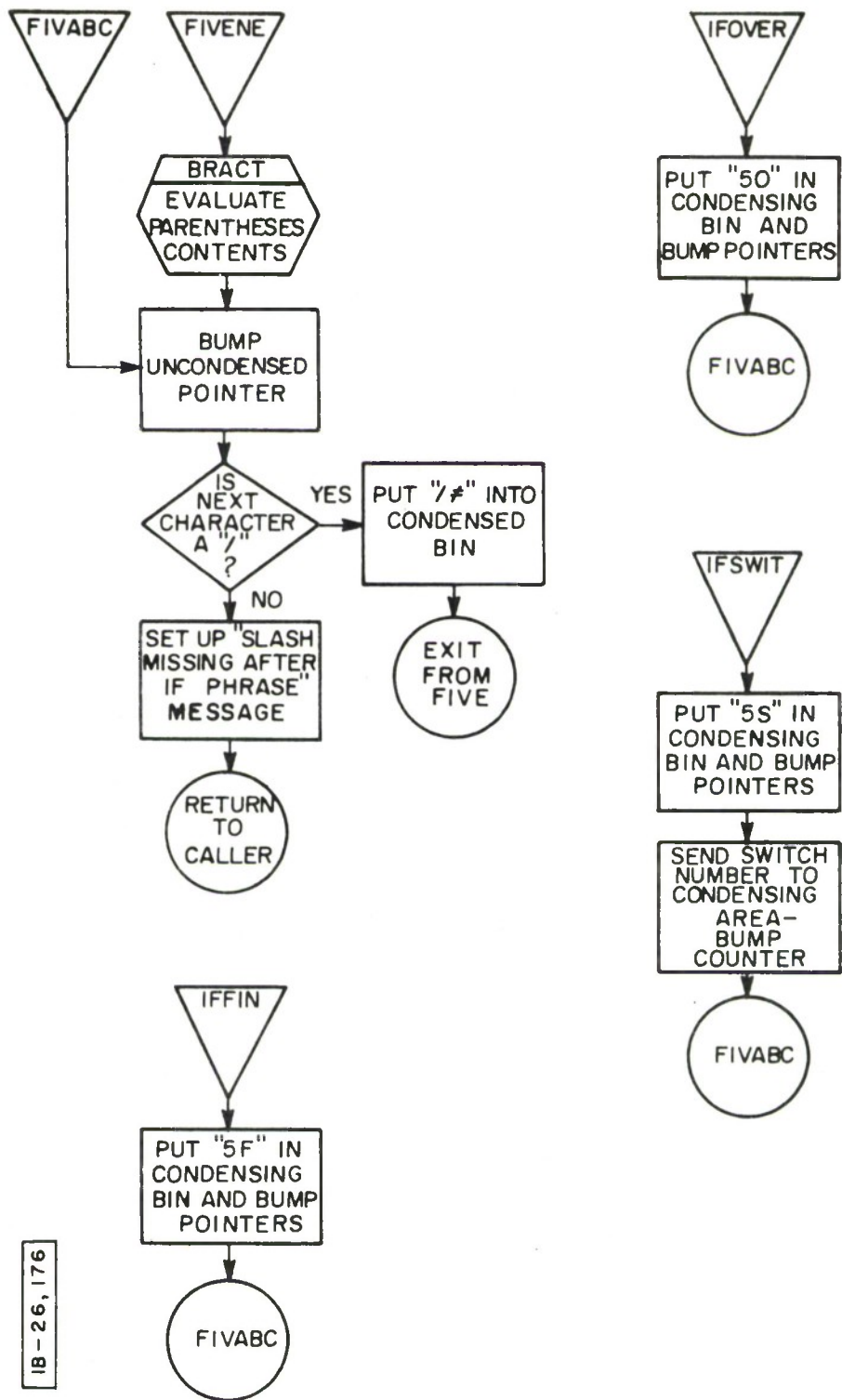
1B-26,259



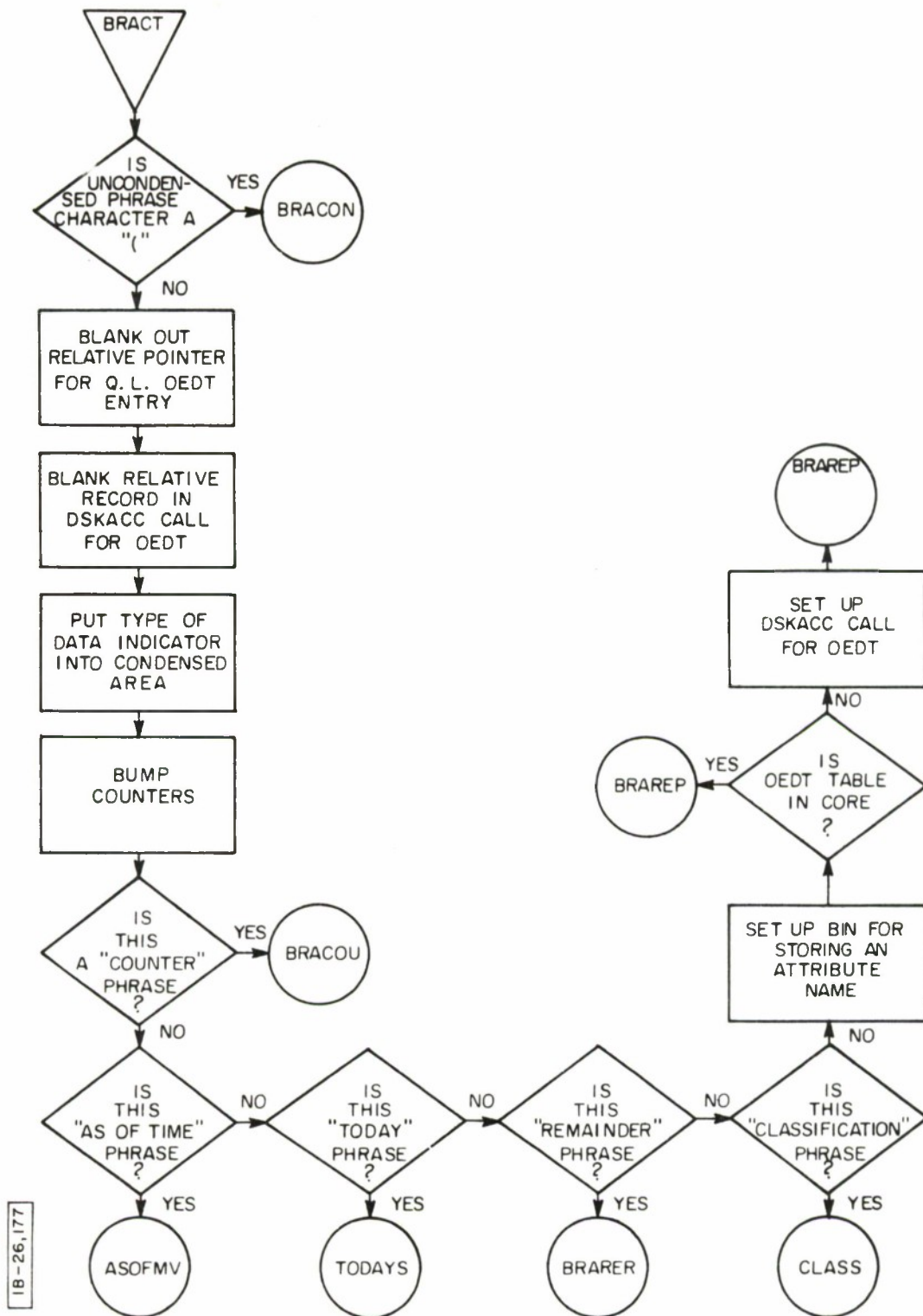
18-26,227

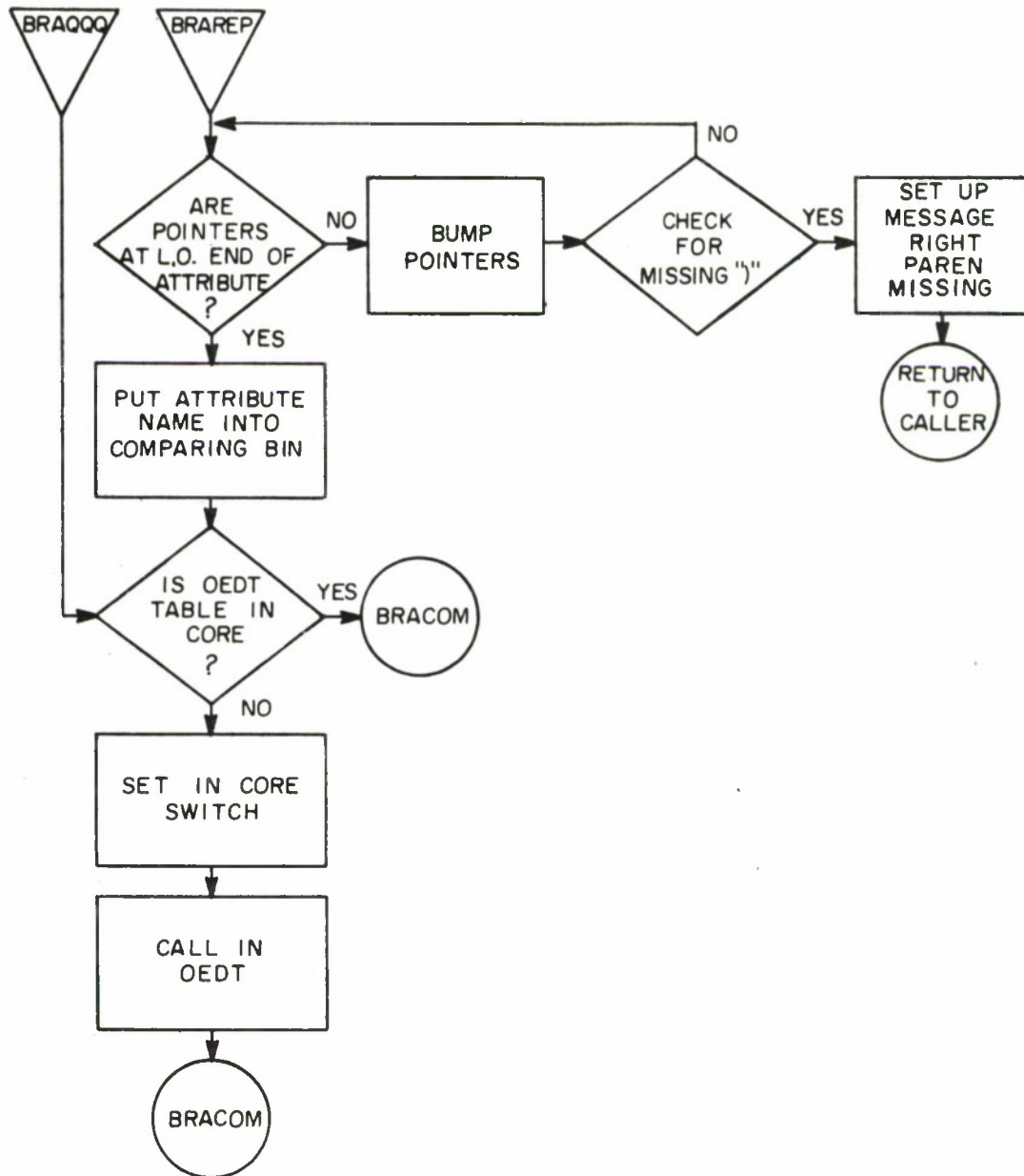


1B-26,228

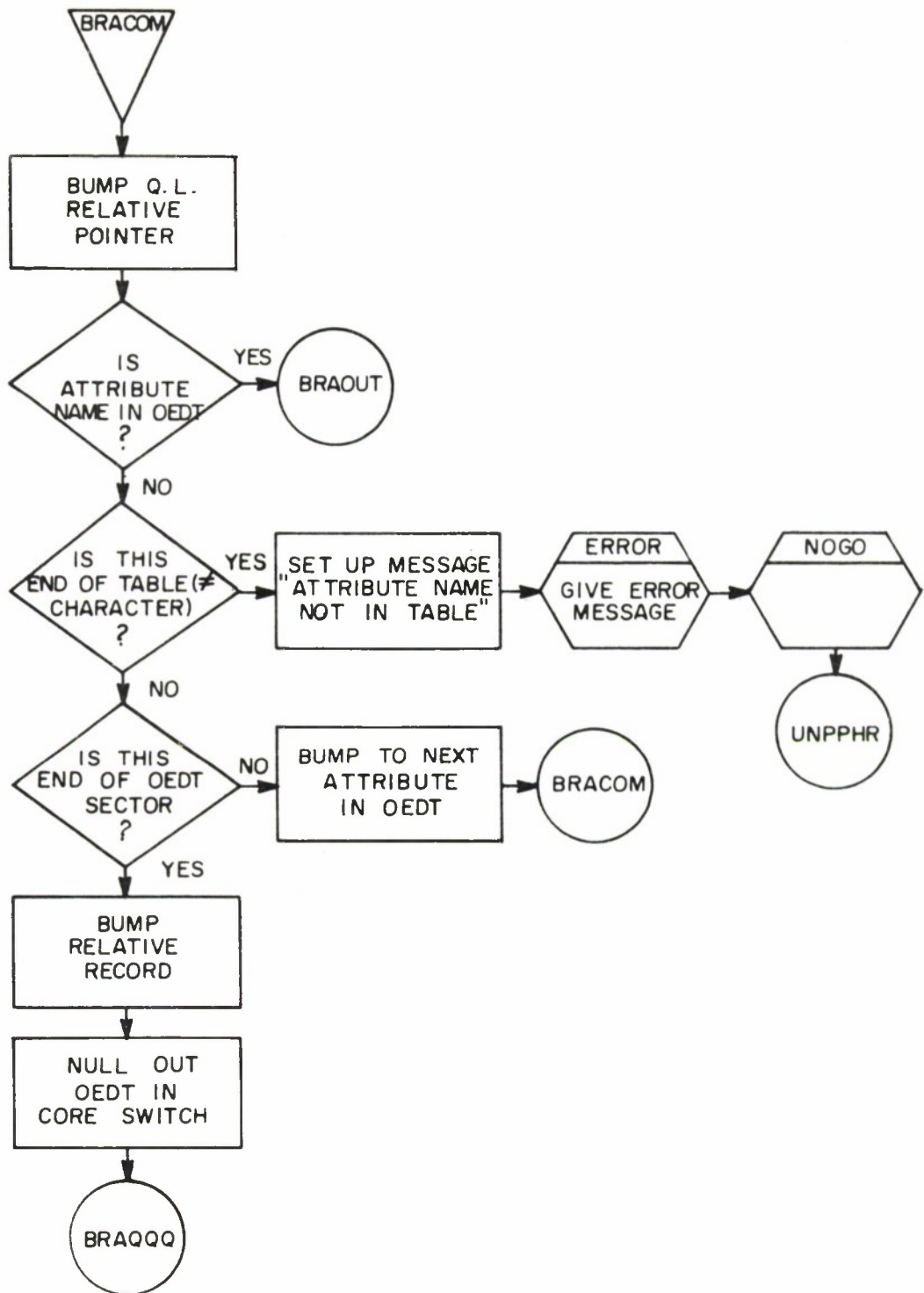


18-26, 176

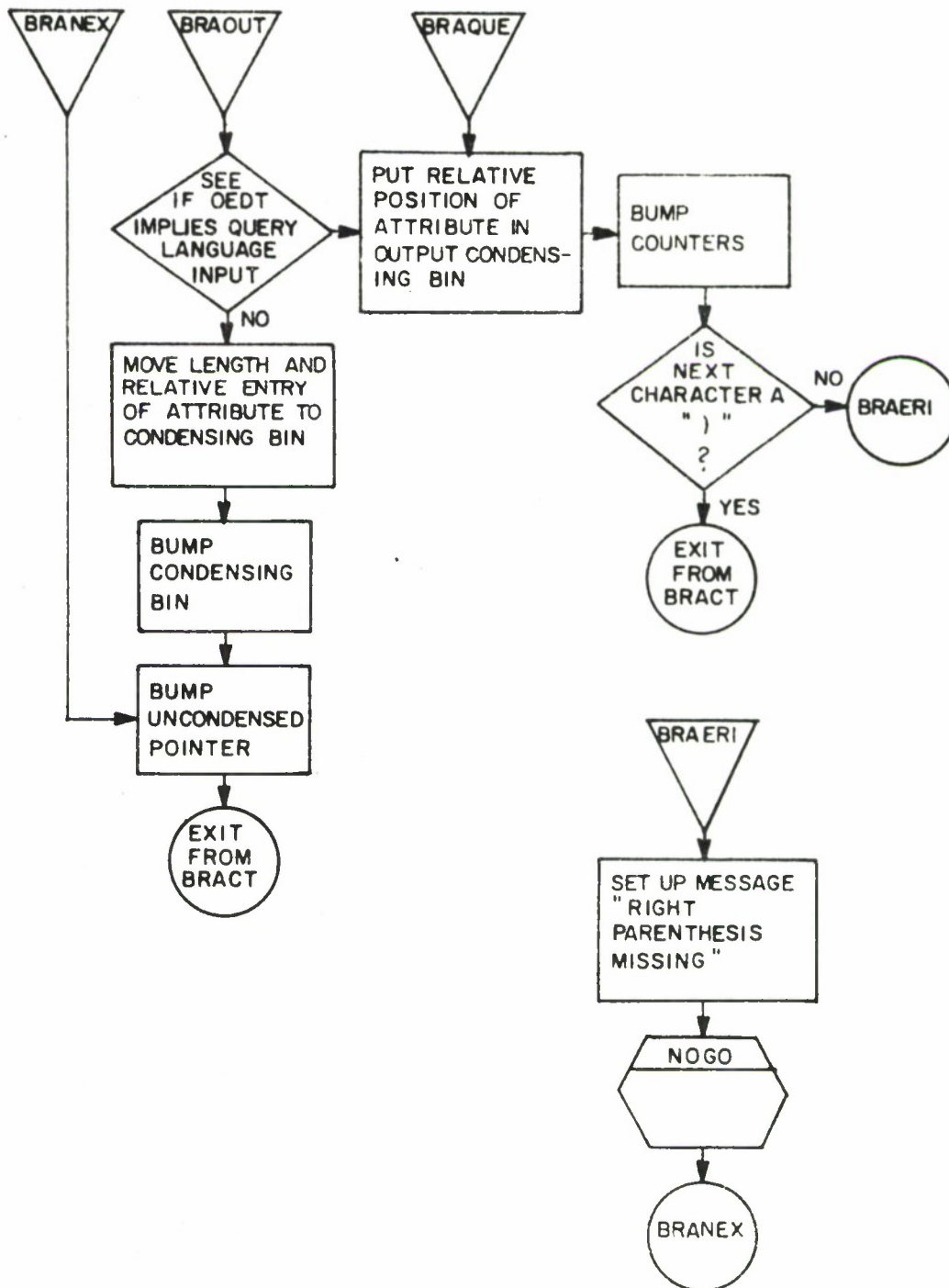




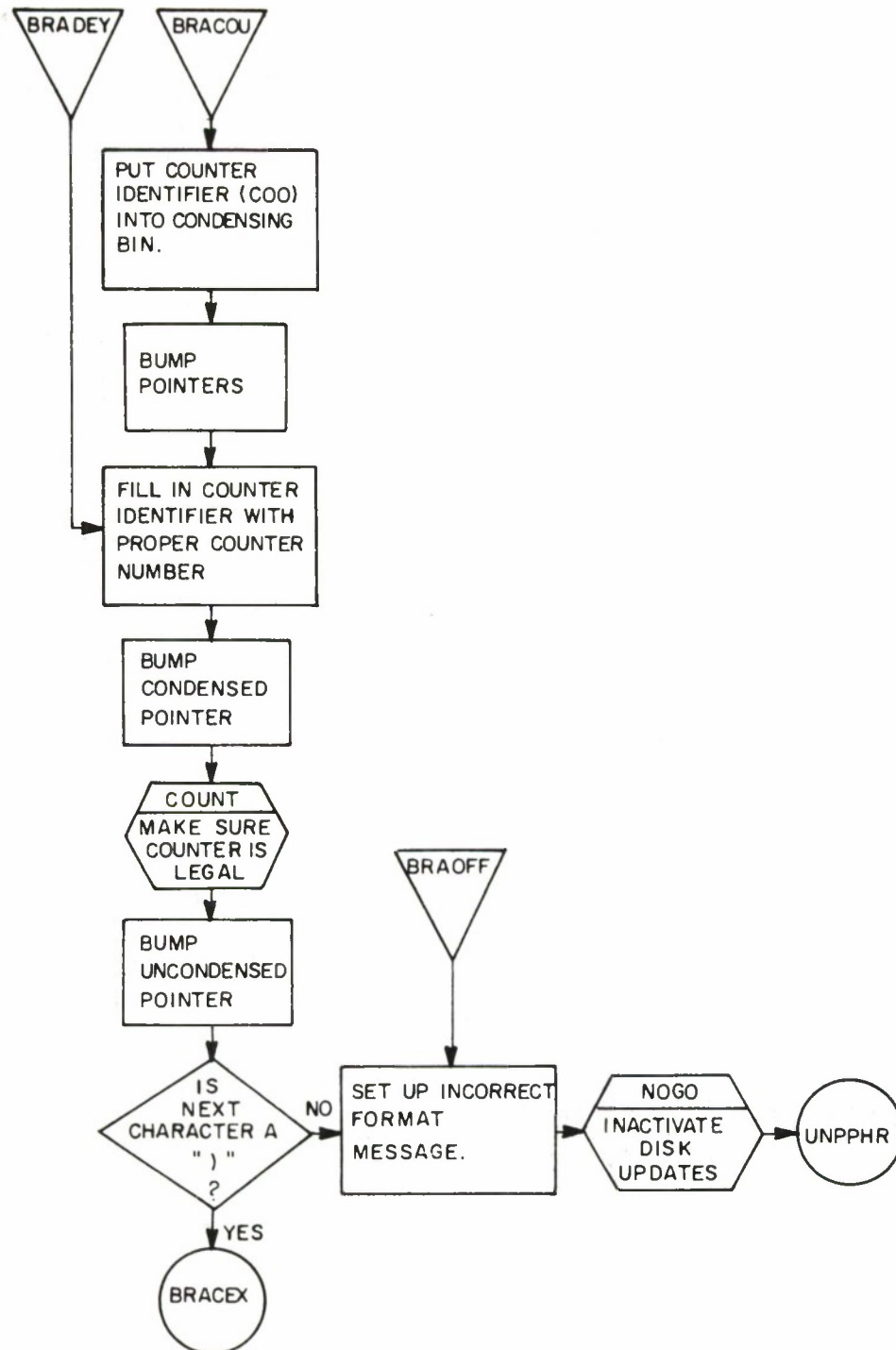
IB-26,215



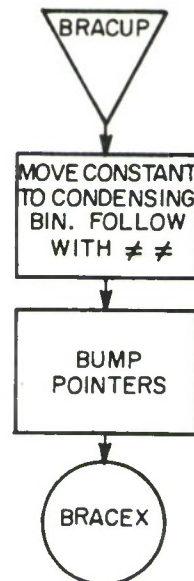
IB - 26, 216



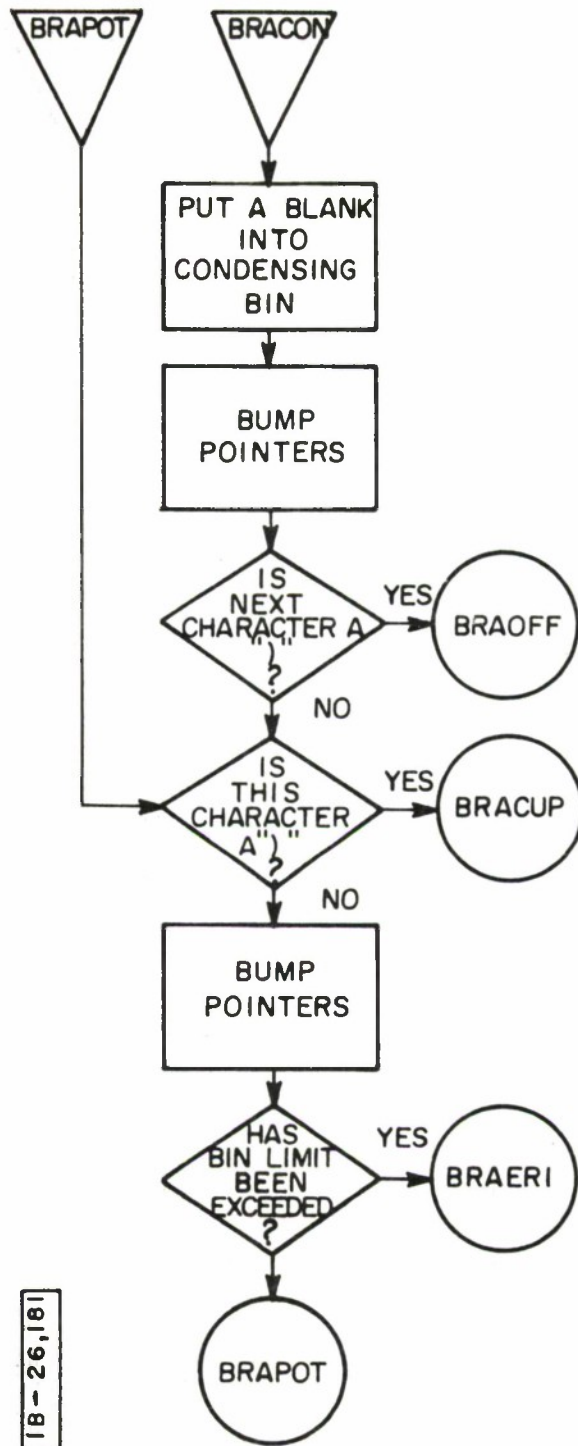
18-26,260



IB - 26, 26



IB - 26,180



18-26,181

REPGEN

Purpose

REPGEN is operated to produce a Serial File Tape from a source data file contained on cards, magnetic tape or on the system disk. The Serial File Tape is then used as input to the output phase of the Report Generation capability.

Calling Sequence

Enter on the console typewriter

REPGENrrrfcn

where rrr = 3-digit report number (no blanks)

f = Format number of the Output Program to be executed.
Must be 1-7 or blank.

c = Classification code: U, C, S or T for Unclassified,
Confidential, Secret or Top Secret.

n = Number of copies of printed output. Must be 1-9 or blank.

Final return will be made to IDLOOP.

Description

The Report Descriptor Table is read from disk and the data from the typewriter call is stored away. If no report number was given with the call, the PRTIND routine is operated, printing an index of REPGEN reports followed by a return to IDLOOP.

If the report number was not blank, the requested report is located in the Report Descriptor Table, mount messages are issued and the input medium is checked.

If the input medium is disk the saved Query Language statement number is taken from the Report Descriptor Table entry and used to construct a dummy call to REPEAT. REPEAT is called via COPS; it fetches the Query Language statement, types it on the console and passes control to GETTAC. GETTAC executes the Query Language statement, placing the retrieved data on a Serial File Tape. The option is now given to the operator to enter the output phase or to terminate.

If the input medium is not disk, the Record Descriptor Table is brought in from disk and a source data record (either card or tape) is read. Entries in the Record Descriptor Table are examined

against the input record until the record type is identified, at which time the Attribute Descriptor Table for this record type is read into core.

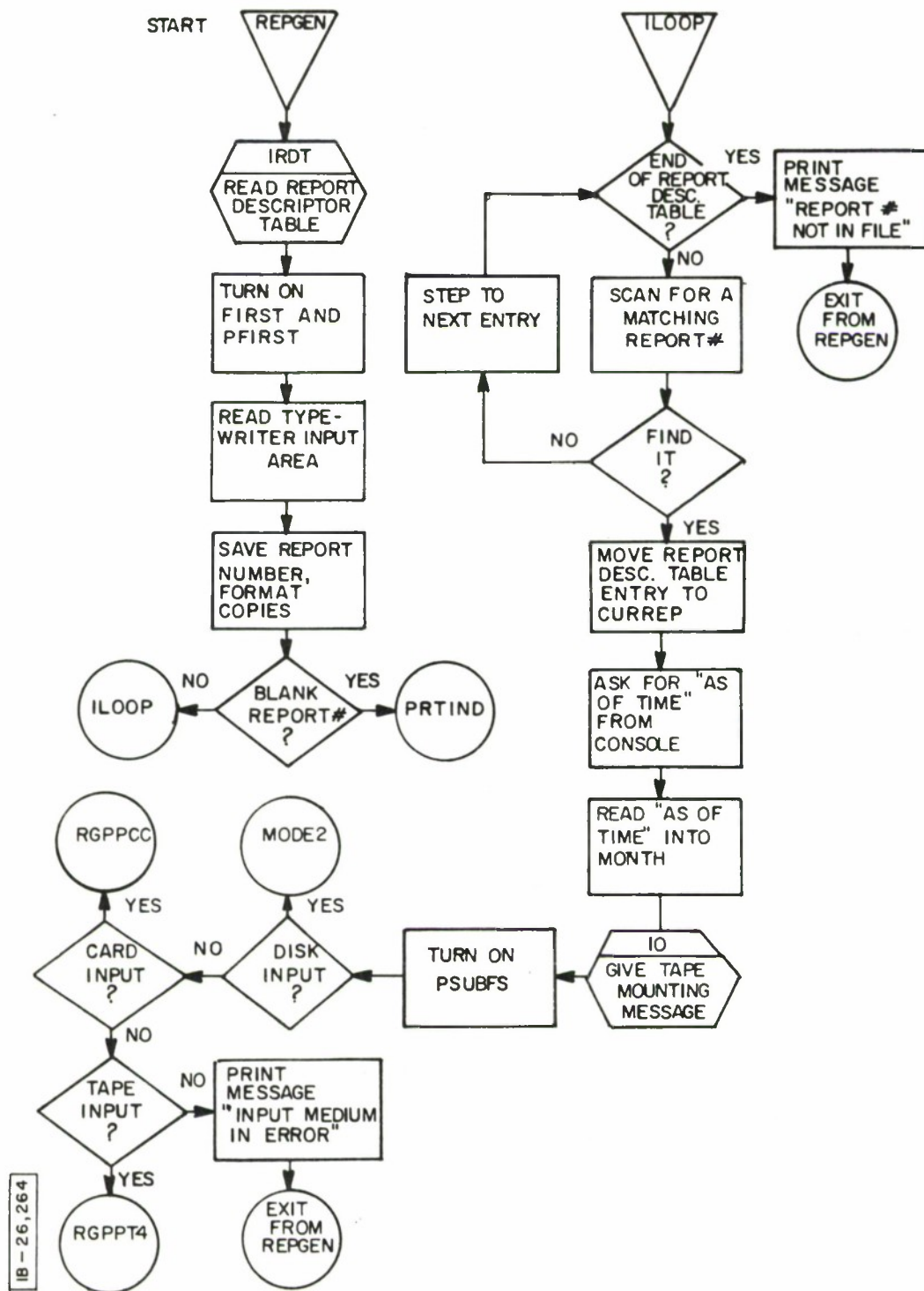
All control attributes are collected into a control field for this input record. Sub-control attributes are collected into a sub-control field.

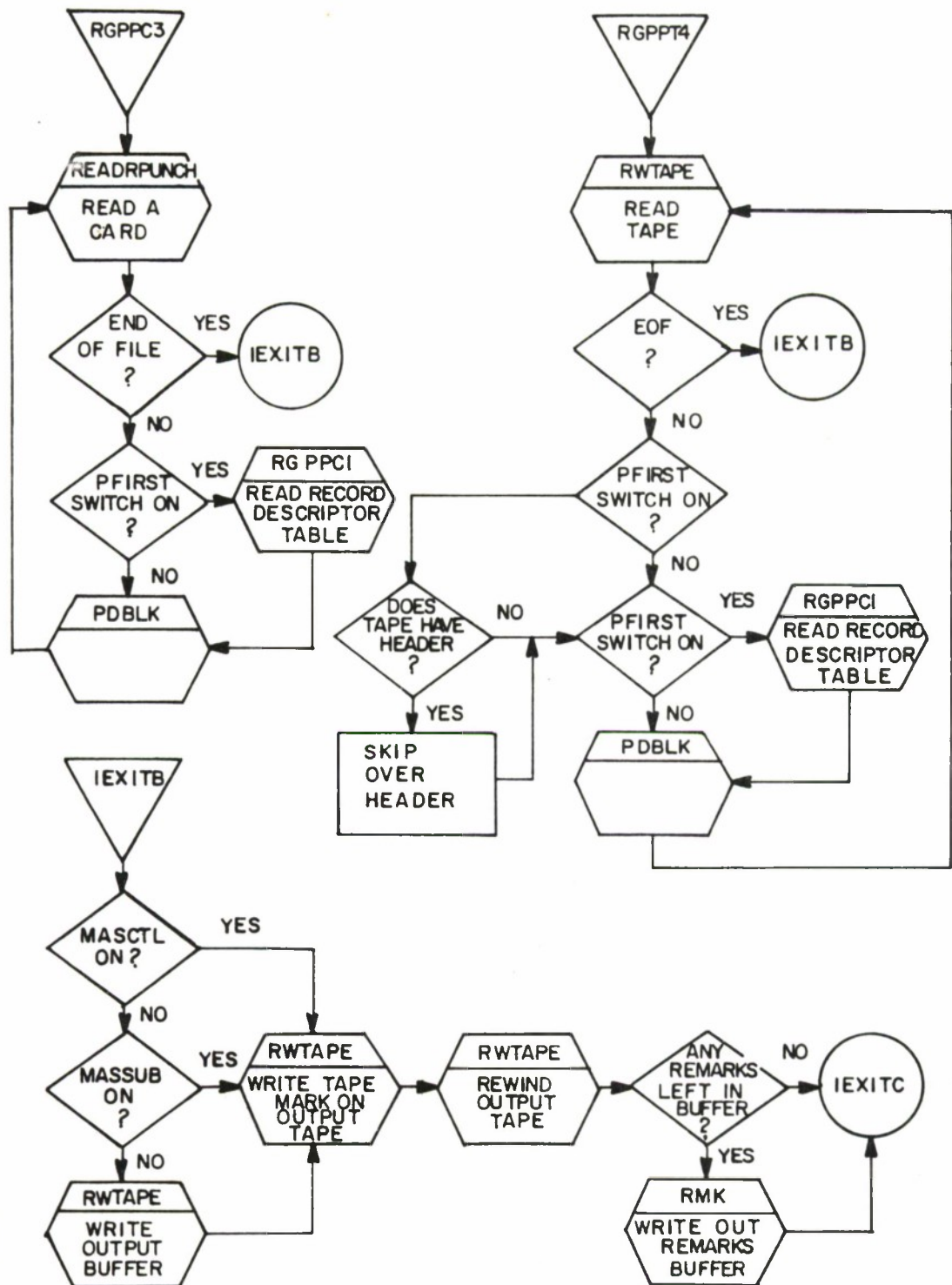
The control field is compared with the previous control field. If they differ, a control break is noted. If the sub-control field differs from the previous sub-control field, a sub-control break is noted. If the current record caused no control or sub-control break, the record is given at once to the audit routine.

The audit routine processes each attribute of the input record separately, performing left or right justification and legality checks as required or specified. Each attribute, after it is audited, results in an entry in the DAWORK Table. The entry in DAWORK specifies the location of the attribute in the input area, its destination in the output area, and its length. If the input record contained sub-control attributes, a similar entry is made in the DAWOR2 table. After all input attributes for the current record have been audited, the DAWORK table is used to move the audited fields from the input area to the output buffer. The next input record is then read.

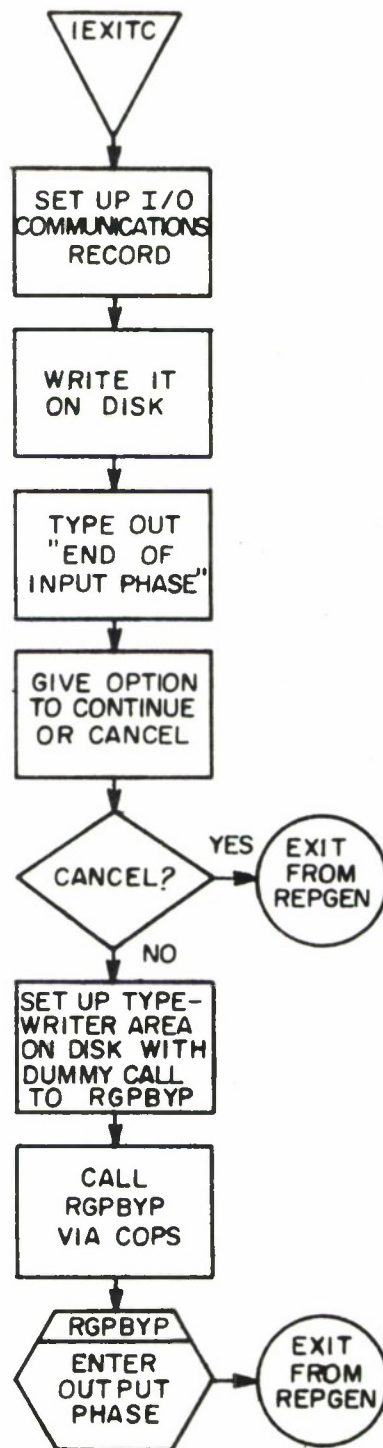
At the point where control and sub-control fields were checked with previous fields, if a control break or sub-control break was noted the current contents of the output buffer will be written on the Serial File Tape. If a control break occurred, the entire output image is then cleared. If a sub-control break occurred, the DAWOR2 table is used to determine those fields to selectively clear in the output image. After the output image is cleared, the current input record is given to the audit routine, and processing continues as described before.

When the input data file has been fully processed and the Serial File Tape is complete, a disk record called the I/O Communication Record is written, containing the current Report Descriptor Table entry and various typewriter inputs (e.g., Report Classification Code, Format Number, AS OF TIME) to be passed along to the output phase. The operator is given a typewriter option to cancel or to continue into the output phase. If the output phase is requested, a dummy call is set up to RGPBYP. Control will eventually be returned from the output phase to REPGEN which will exit to IDLOOP.

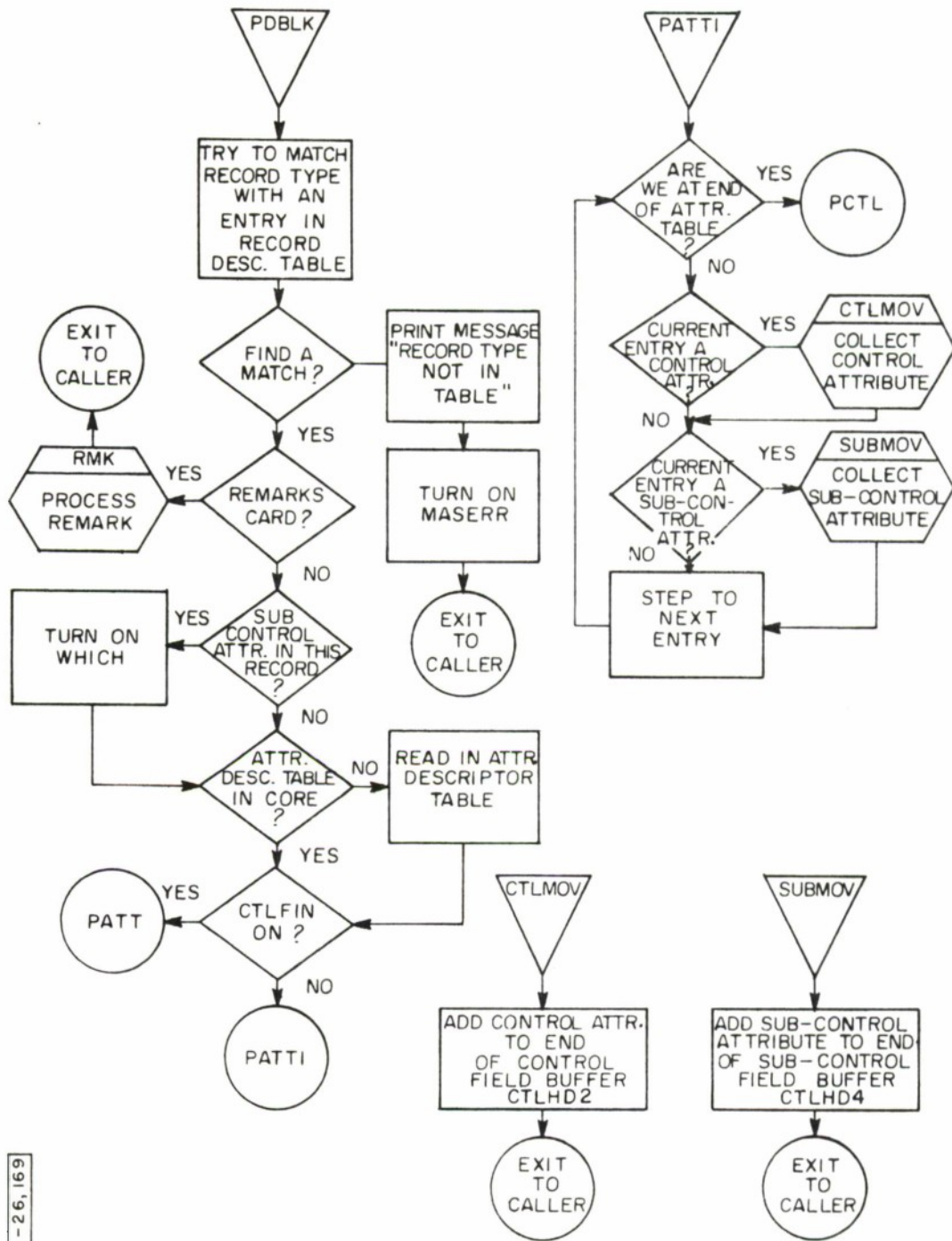




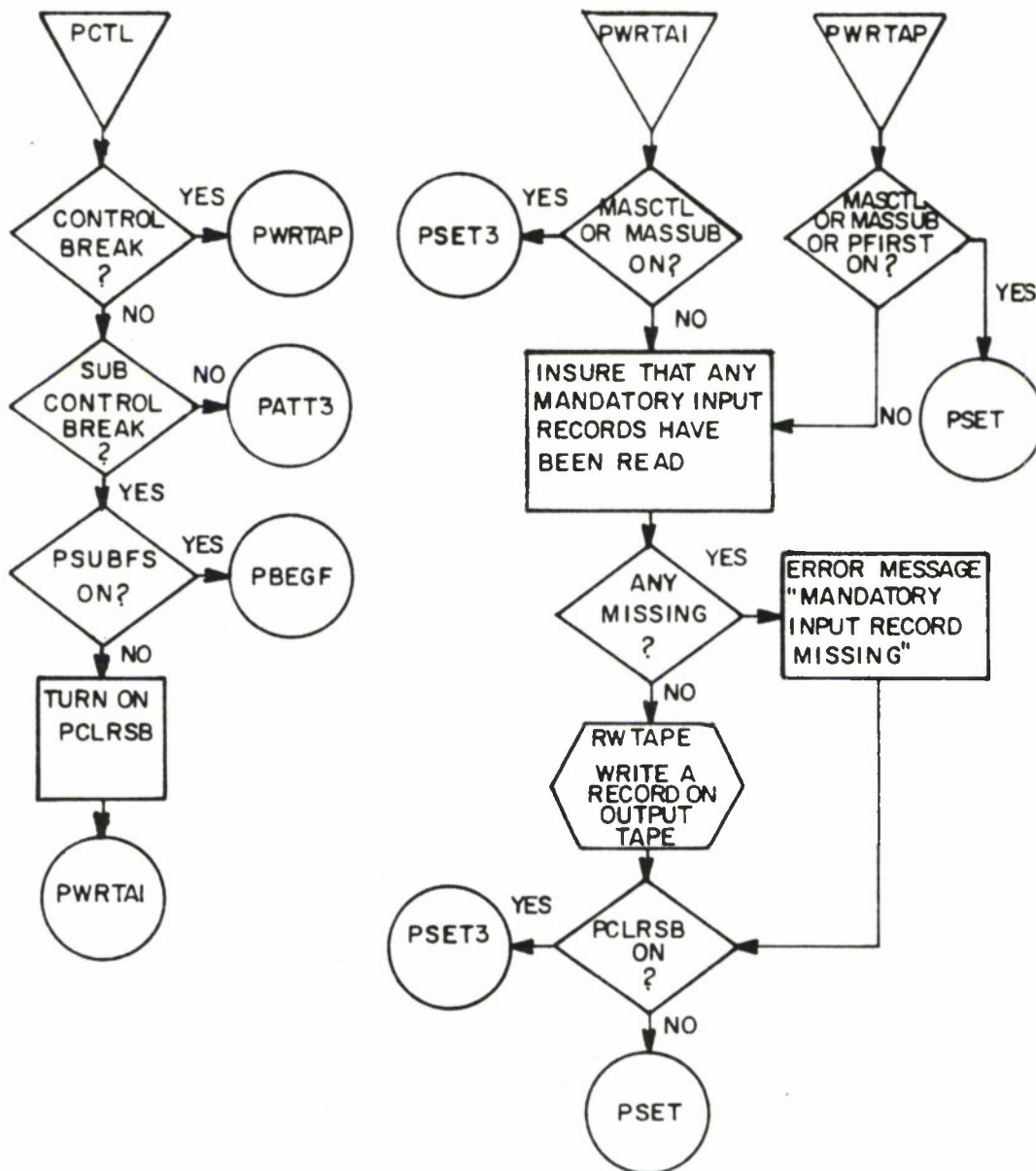
11B-26,265

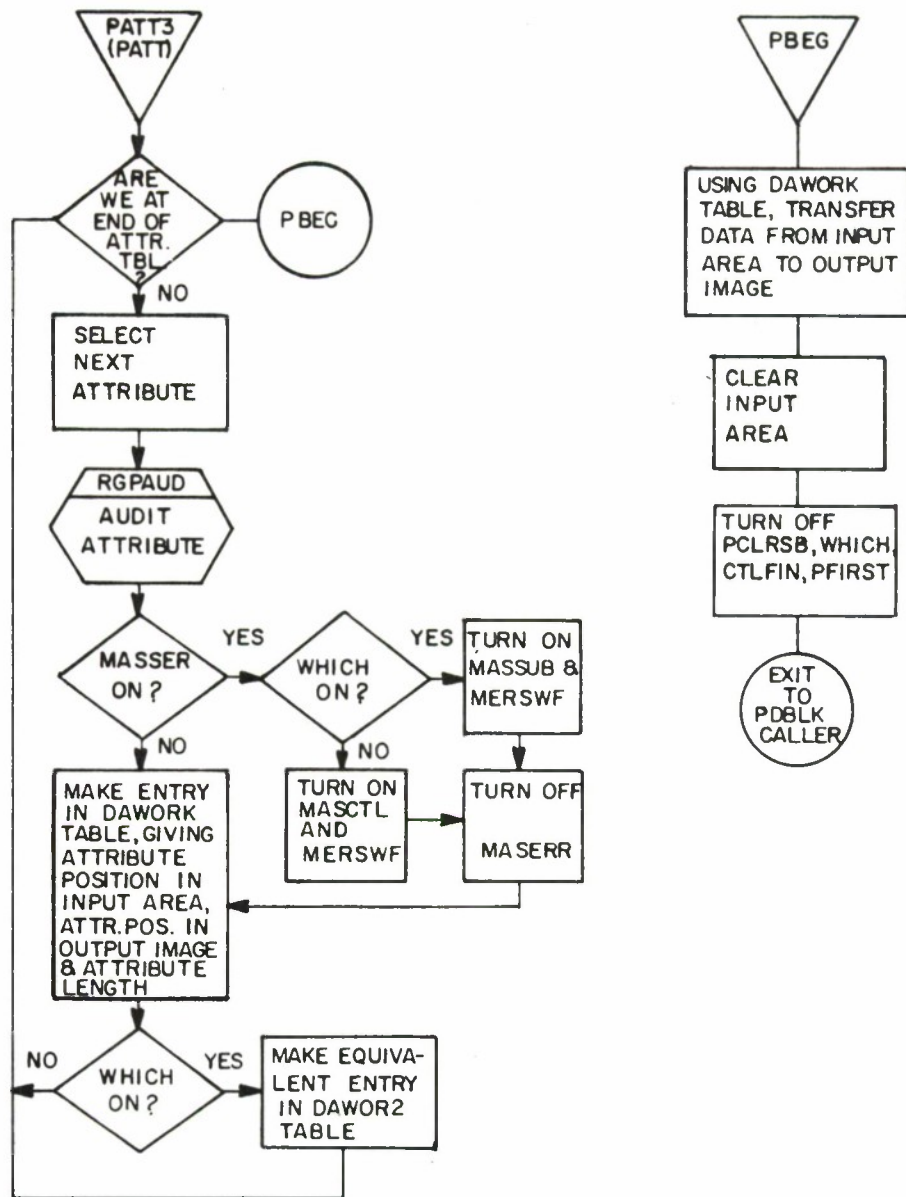


1B - 26, 168

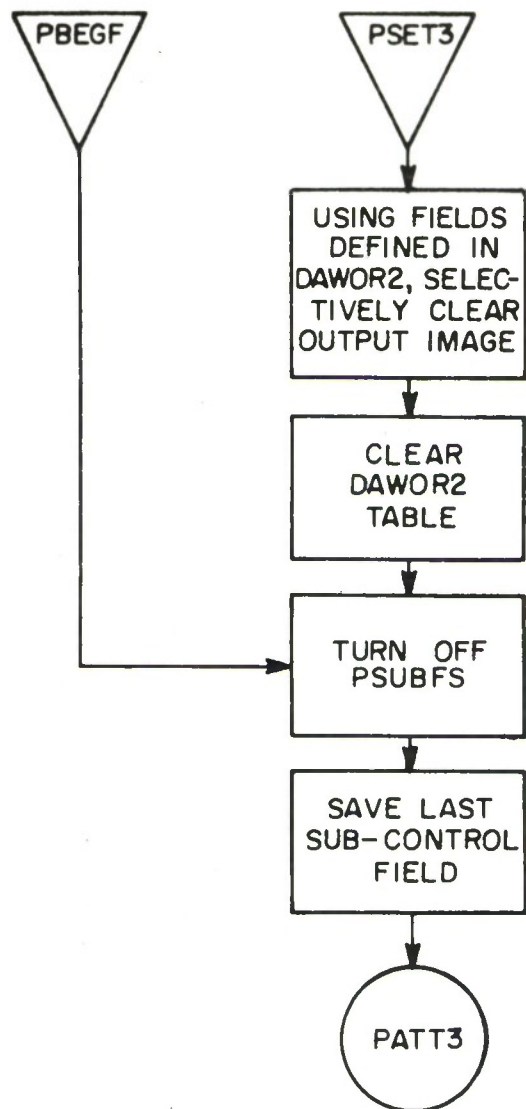
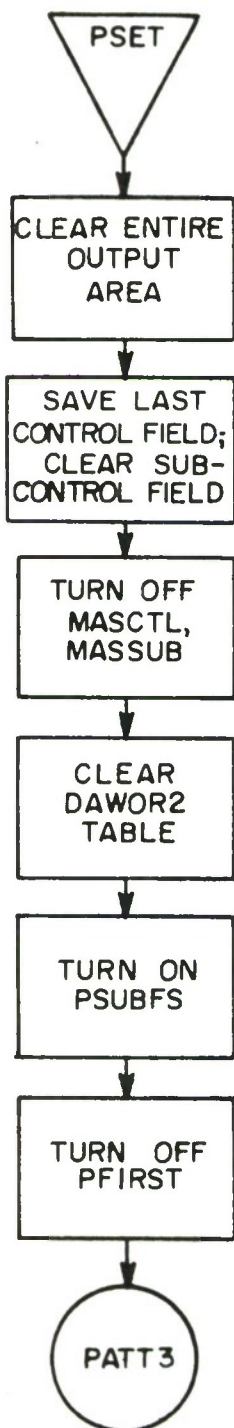


IB - 26, 169

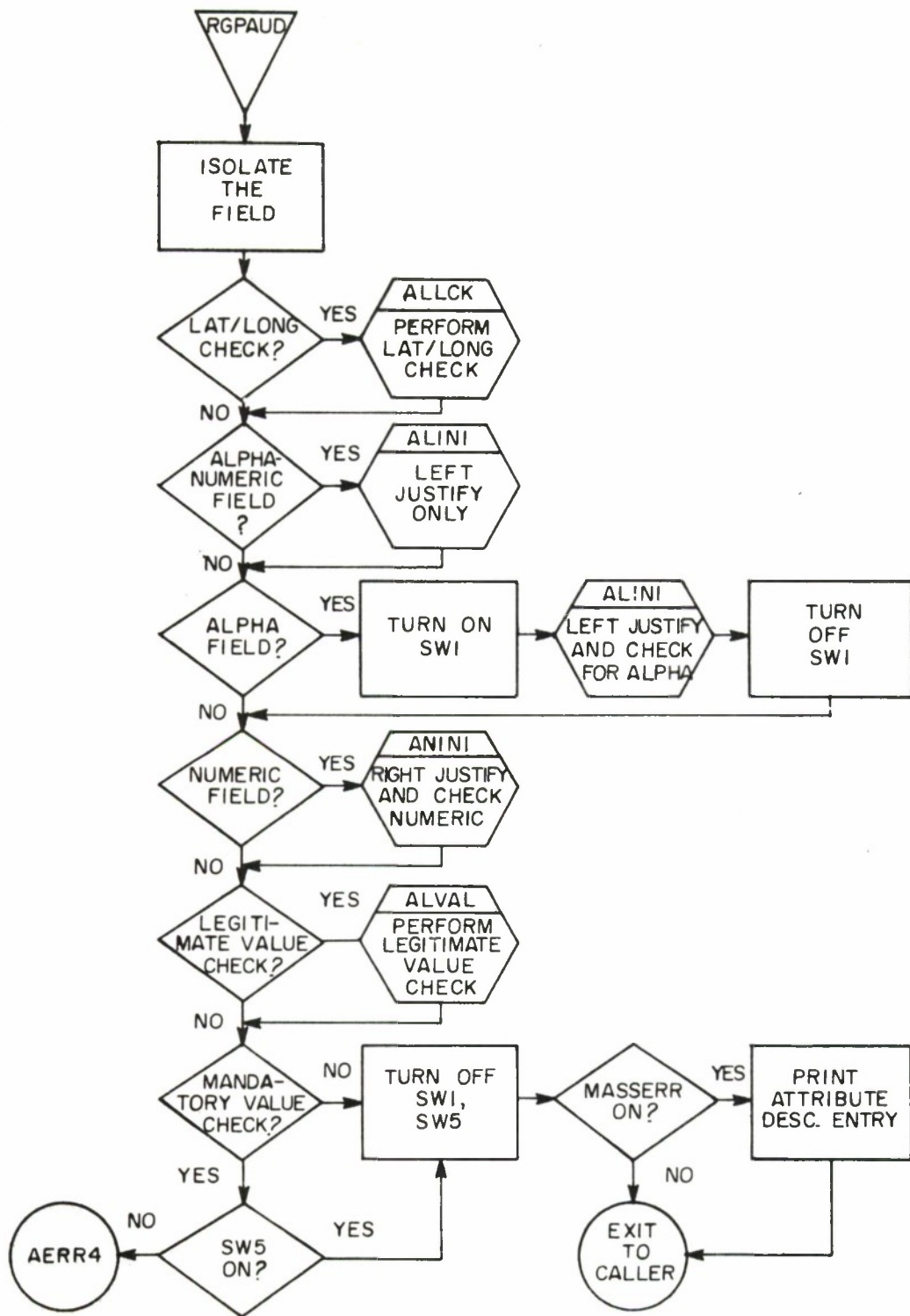


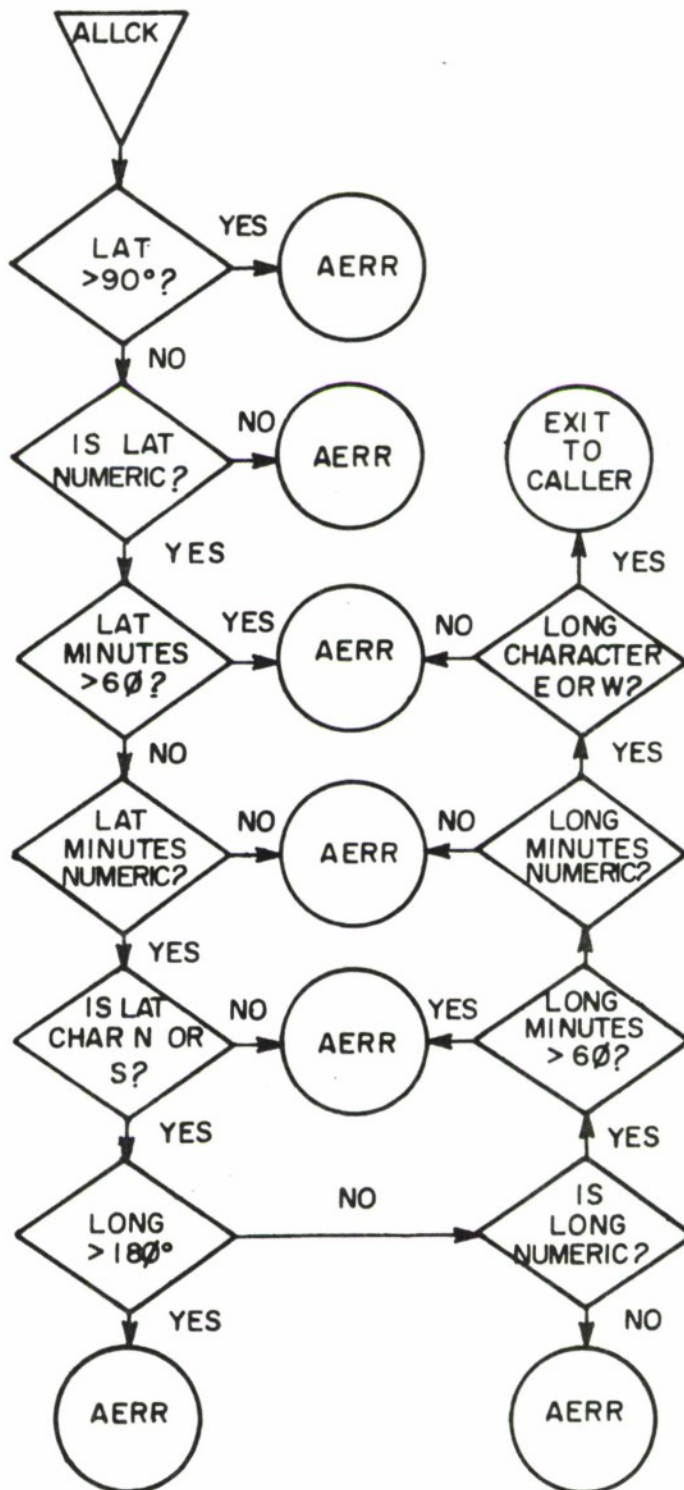


18 - 26, 263

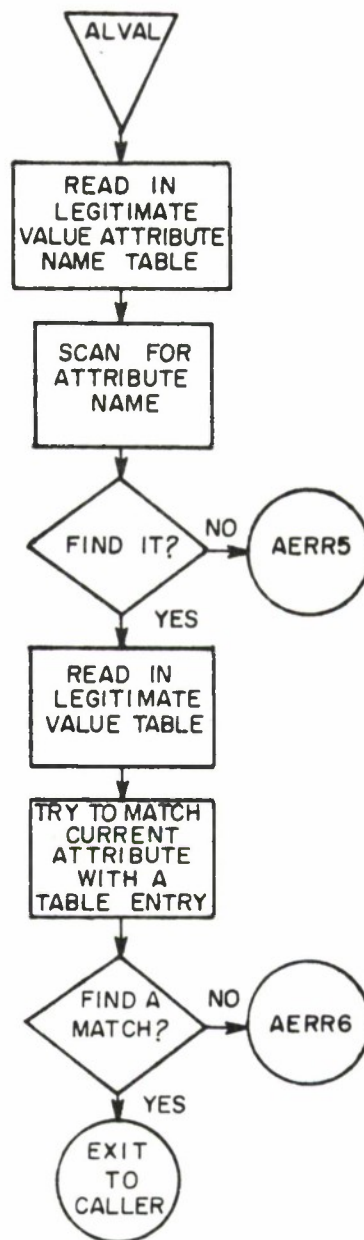
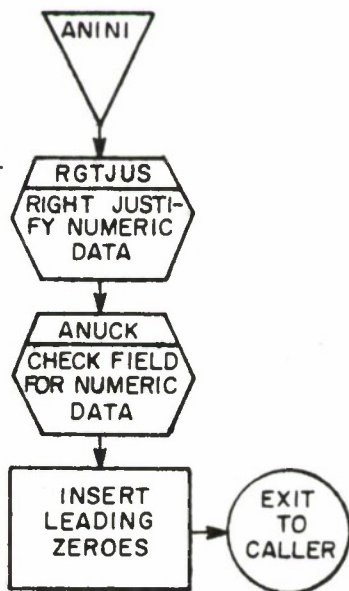
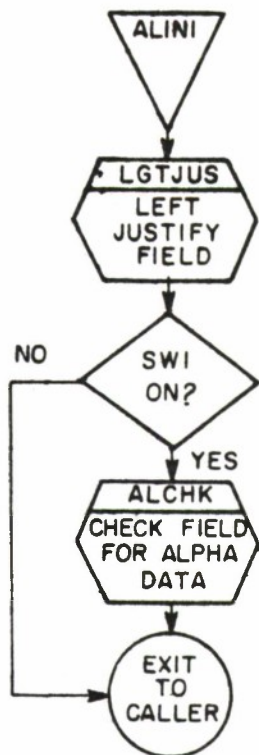


IB-26,171

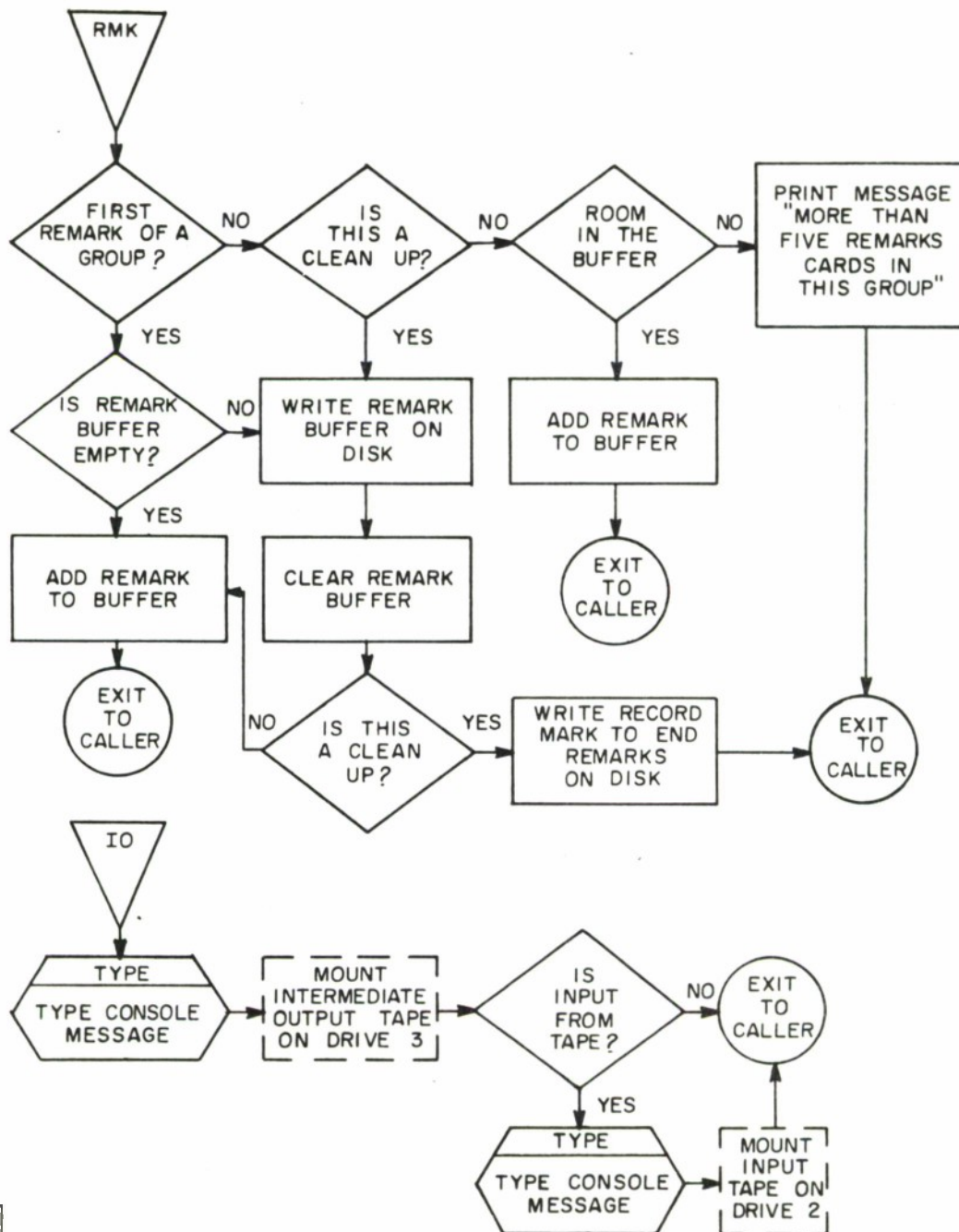




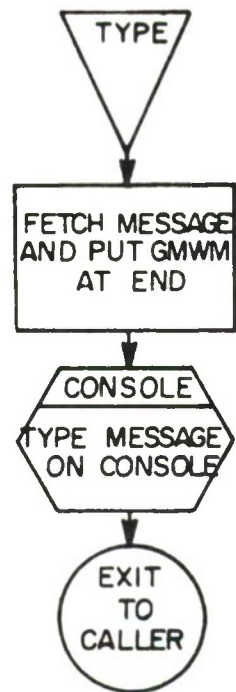
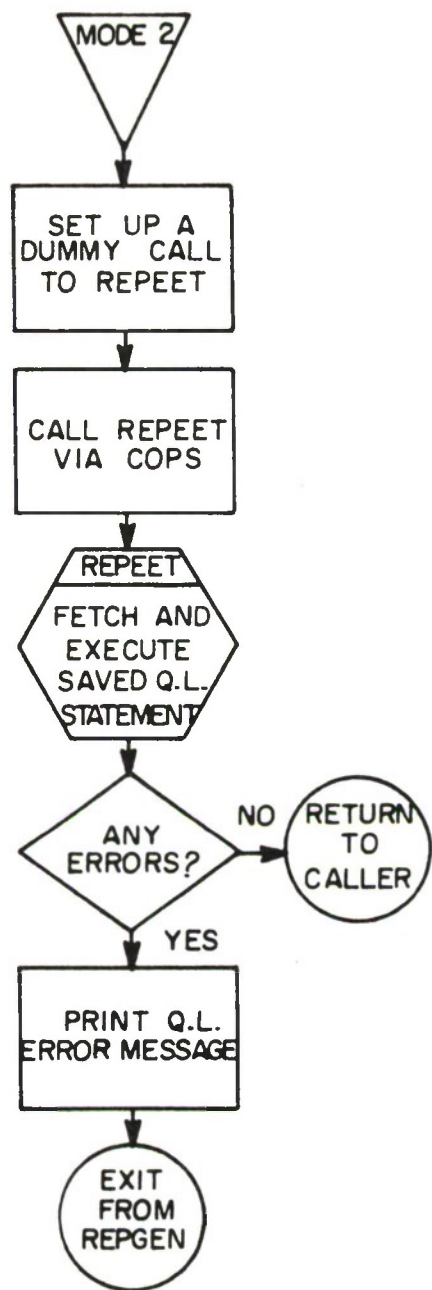
18-26,178



118-26,179



1B-26,222



IB-26,221

RGPBYP

Purpose

This routine issues mount messages for the output phase, completes the I/O Communication Record and calls in the output phase.

Calling Sequence

If called internally from REPGEN, the dummy typewriter call which is constructed is

RGPBYP***

If called externally, enter on the console typewriter

RGPBYPrrrfcn

where rrr = 3-digit report number (no blanks)

 f = Format number of the Output Program to be executed.
 Must be 1-7 or blank.

 c = Classification code: U, C, S or T for Unclassified,
 Confidential, Secret or Top Secret.

 n = Number of copies of printed output. Must be 1-9 or blank.

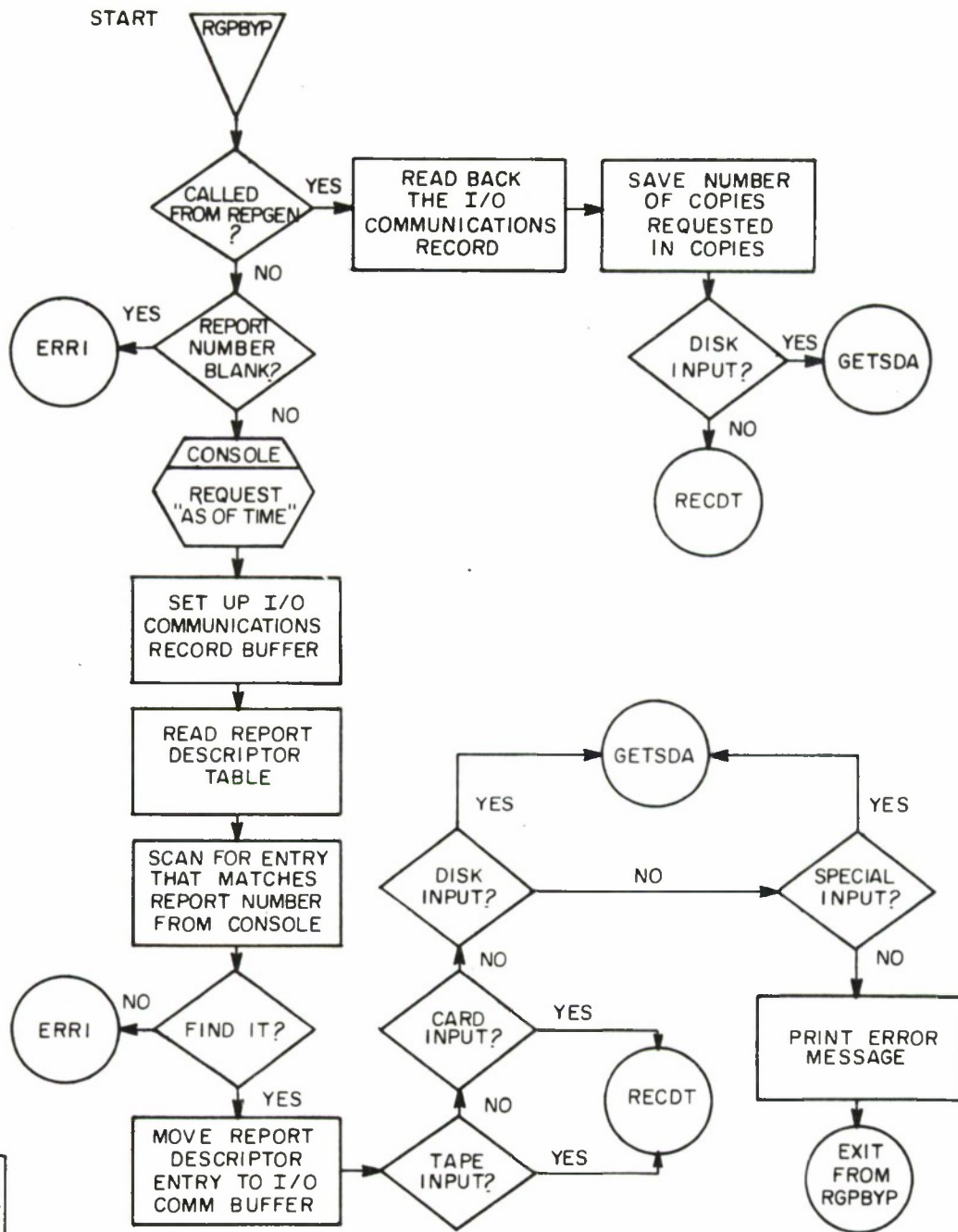
Final return is to calling program.

Description

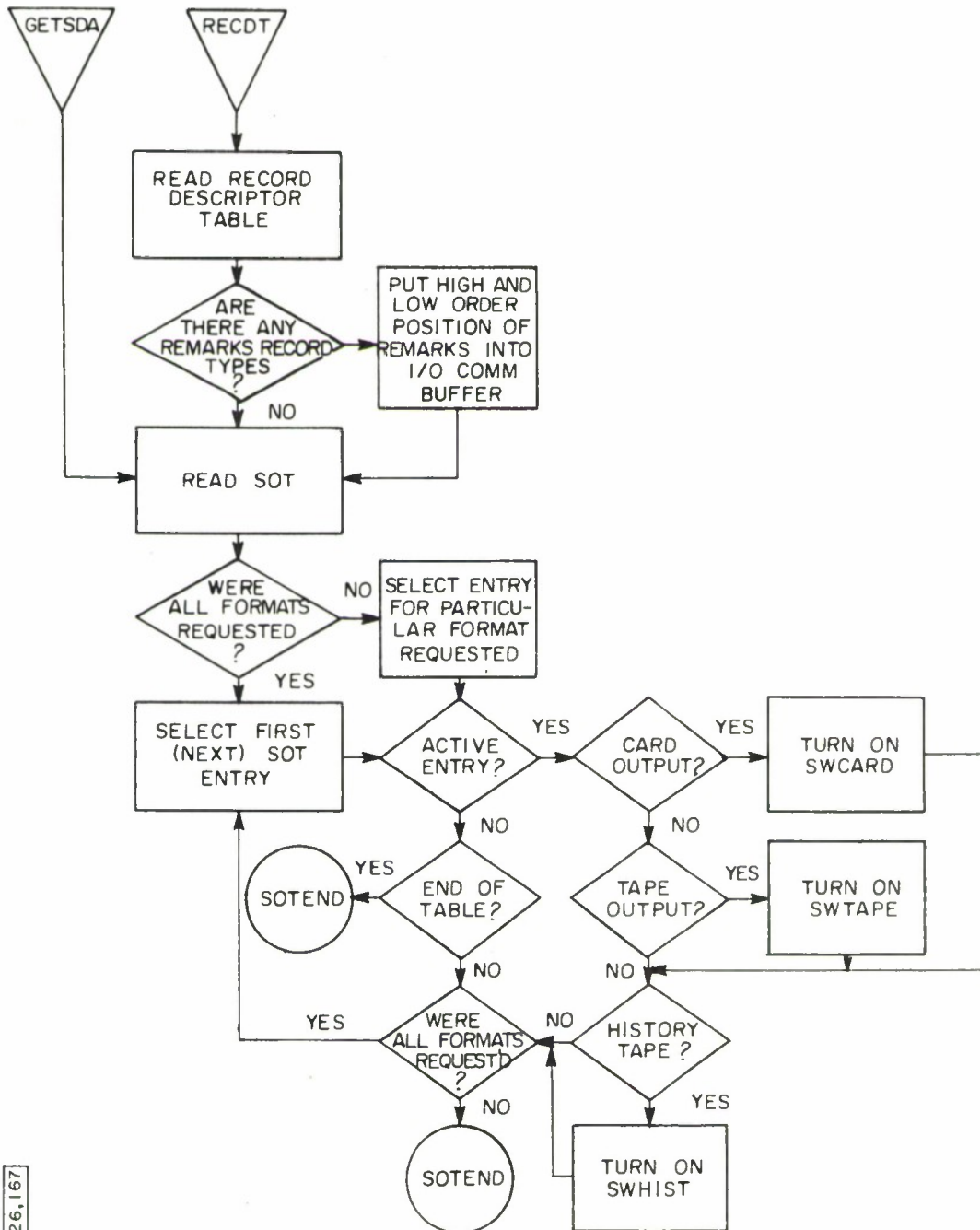
If RGPBYP was called from REPGEN, the I/O Communication Record is read back from disk. If RGPBYP was externally called, the AS OF TIME is requested, and an I/O Communication Record is constructed.

The SOT Table for the requested report is accessed and used to determine the output devices required and the mount messages to be issued. Requests for multiple copies are ignored if the output device is not the printer.

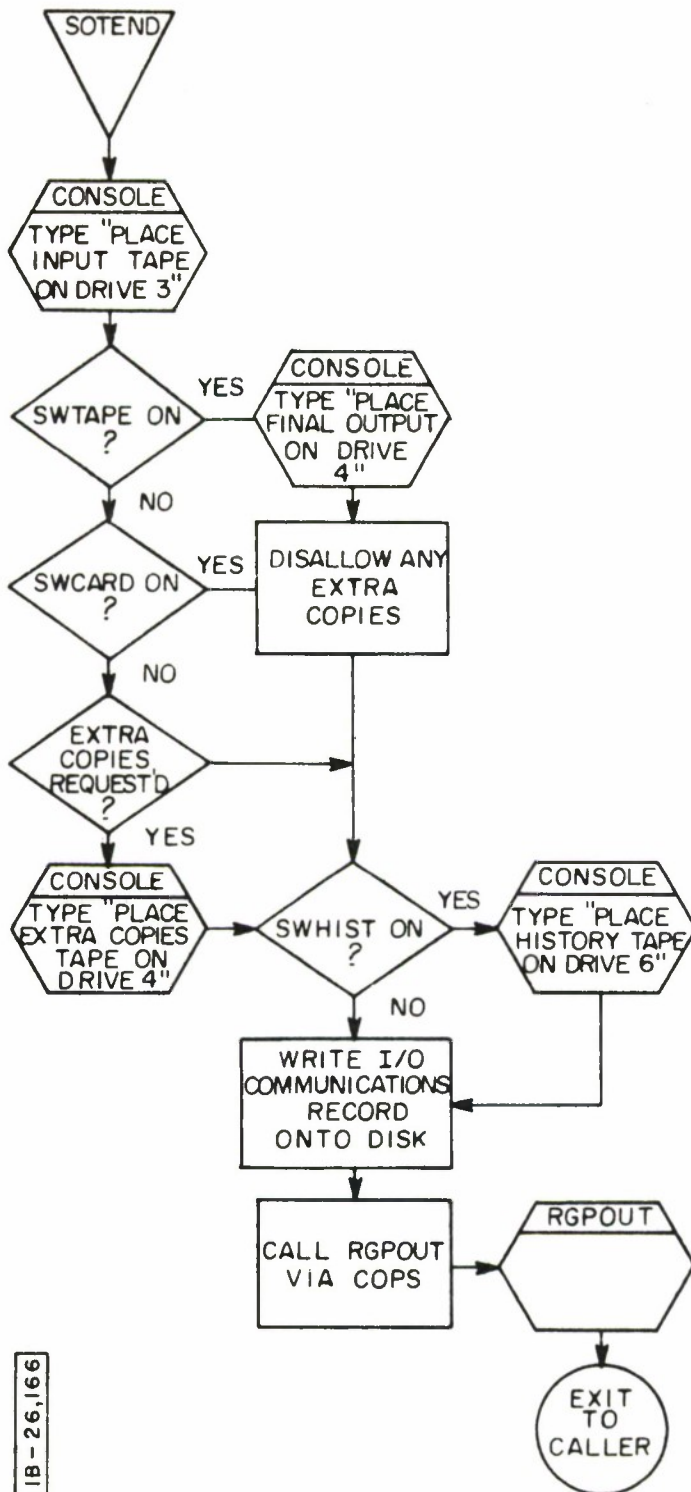
After the mount messages are issued, the I/O Communication Record is completed and written on the system disk. The output program RGPOUT is then called via COPS. After return is made from RGPOUT, RGPBYP will exit to its calling program.



IB-26,210



IB-26,167



IB - 26,166

RGPOUT

Purpose

RGPOUT acts as the output phase supervisor, controlling the loading and execution of the various output phase programs.

Calling Sequence

RGPOUT is called internally from RGPBYP as follows:

```
B    COPS
      DCW 'RGPOUT'
```

Description

The Read/Write Tape routine RWTAPE, the CONSOLE routine and the initial Communication Zone are all assembled with RGPOUT, and are loaded into core when RGPOUT is loaded. They remain in core throughout the output phase.

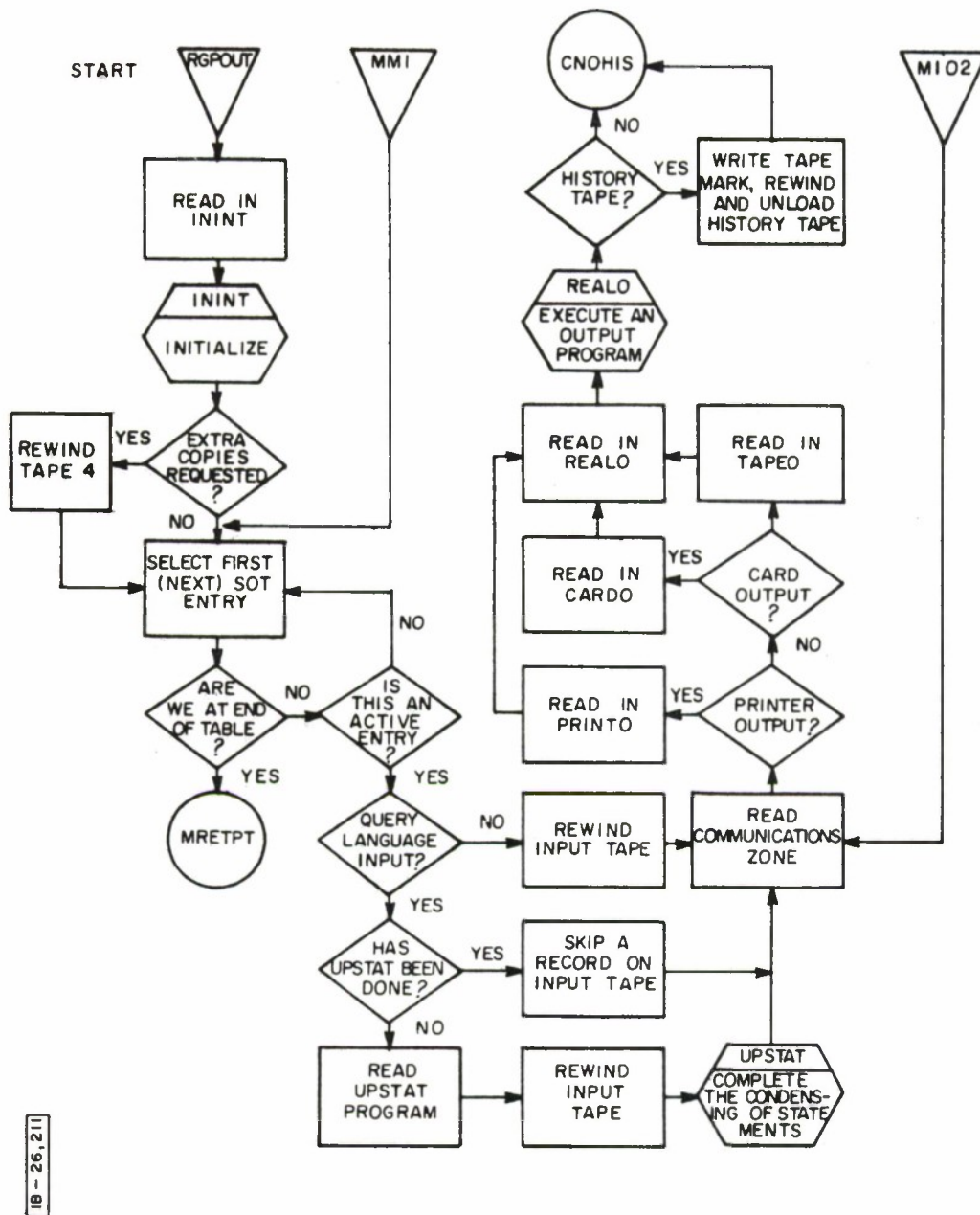
RGPOUT calls the output phase initialization routine ININT into core and executes it. Next the SOT Table entry for the output program to be executed is examined. If input was via Query Language, the SOT entry is checked to see if UPSTAT has been executed for this output program.

If input was via Query Language and UPSTAT has not been executed, UPSTAT is loaded and executed to complete the condensing of the output program.

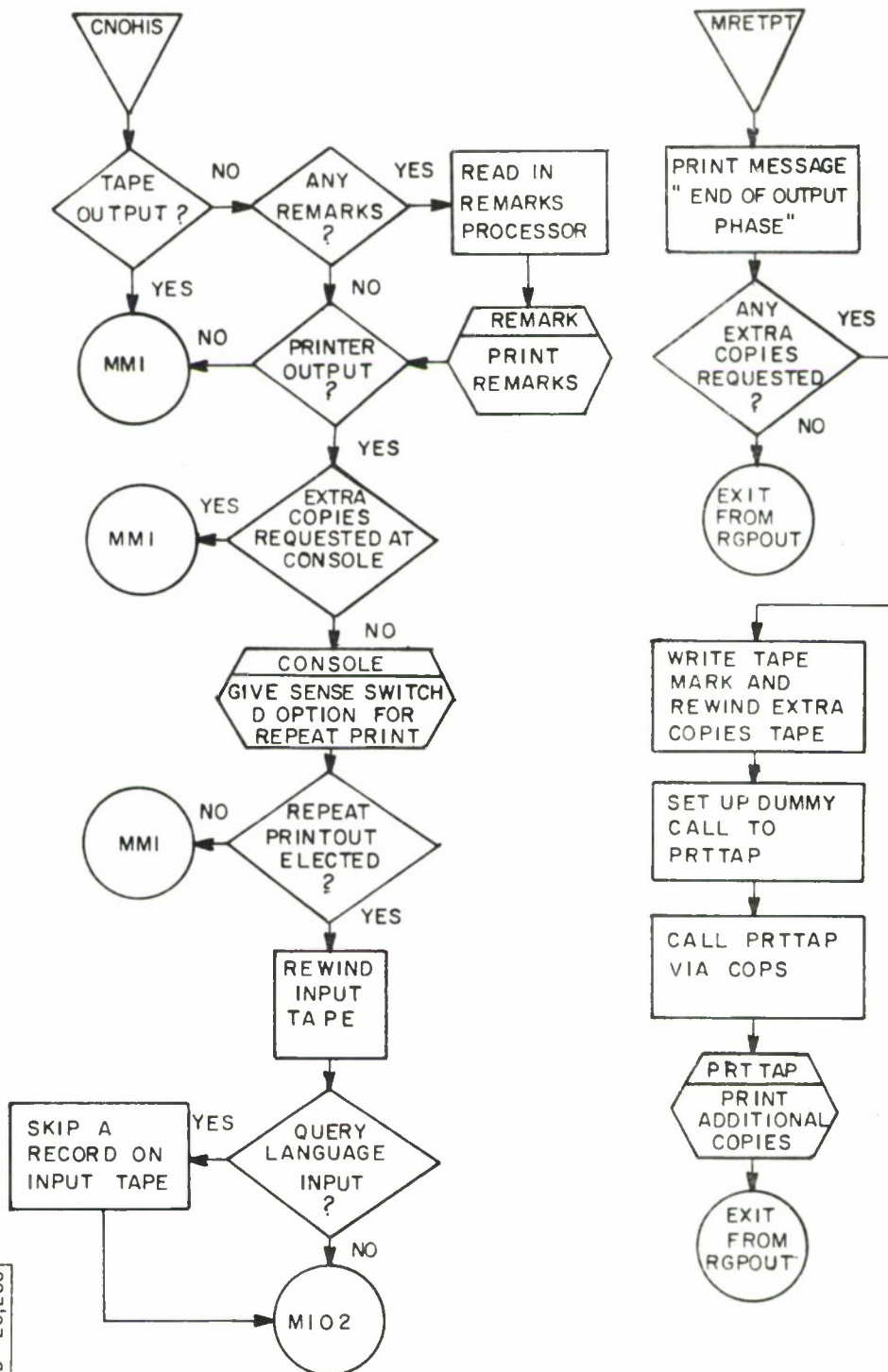
The appropriate output routine, PRINTO, CARDO or TAPEO, is loaded depending upon whether the output device is the printer, the card punch or magnetic tape. The main output program REALO is loaded and executed to produce the desired output.

After REALO executes the output program, RGPOUT checks to see if any more formats were requested for this report. If so, the appropriate output program is loaded, REALO is loaded, and the next output program is executed.

When all requested output programs have been executed, RGPOUT checks to see if extra copies of printed reports were requested. If so, the extra copies tape is rewound and an internal call to PRTTAP is set up and executed to yield the necessary printed copies.



IB - 26, 211



IB-26,266

ININT

Purpose

Performs initialization for the output phase of REPGEN.

Calling Sequence

ININT is called from RGPOUT by

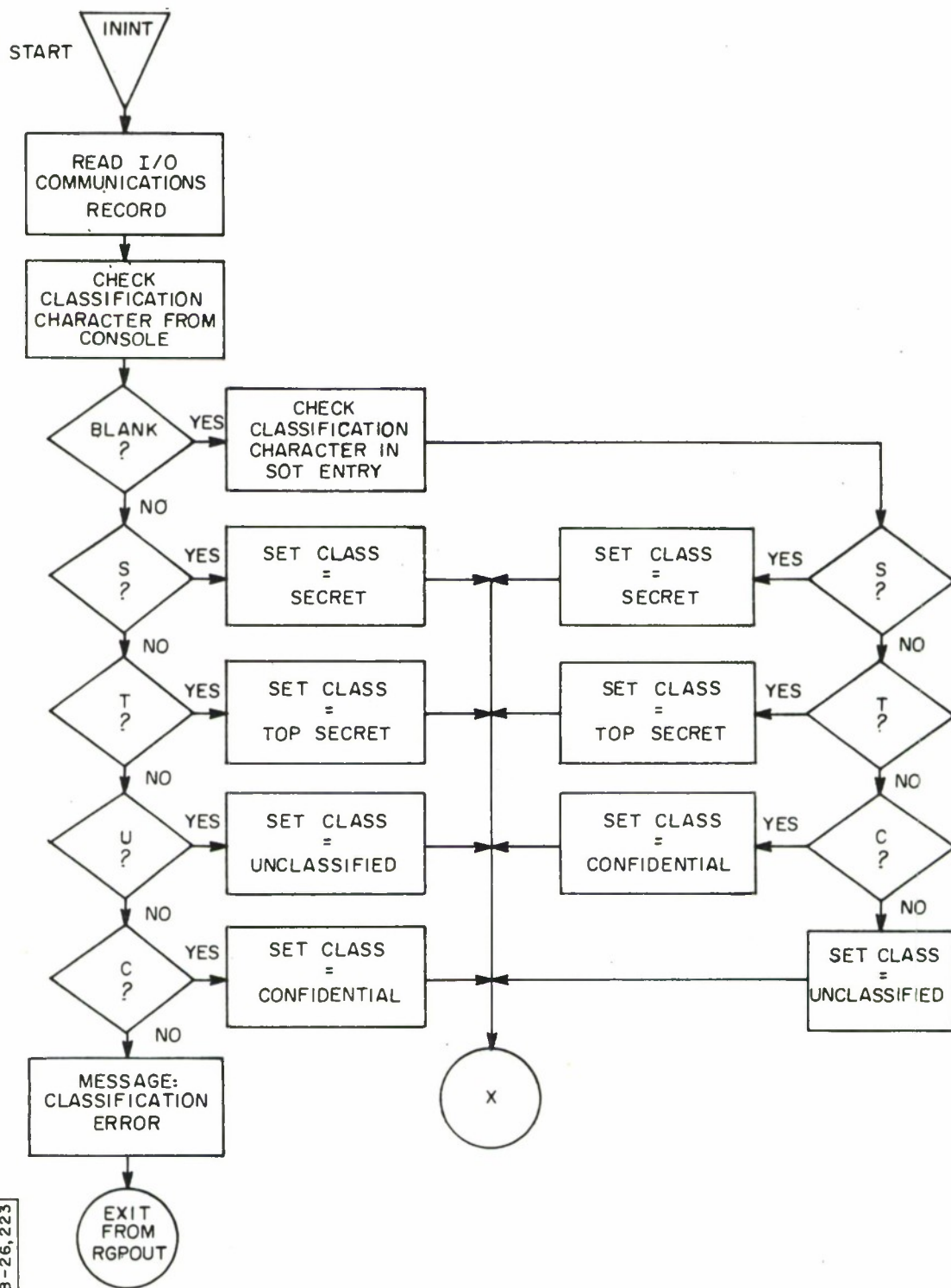
B ININT

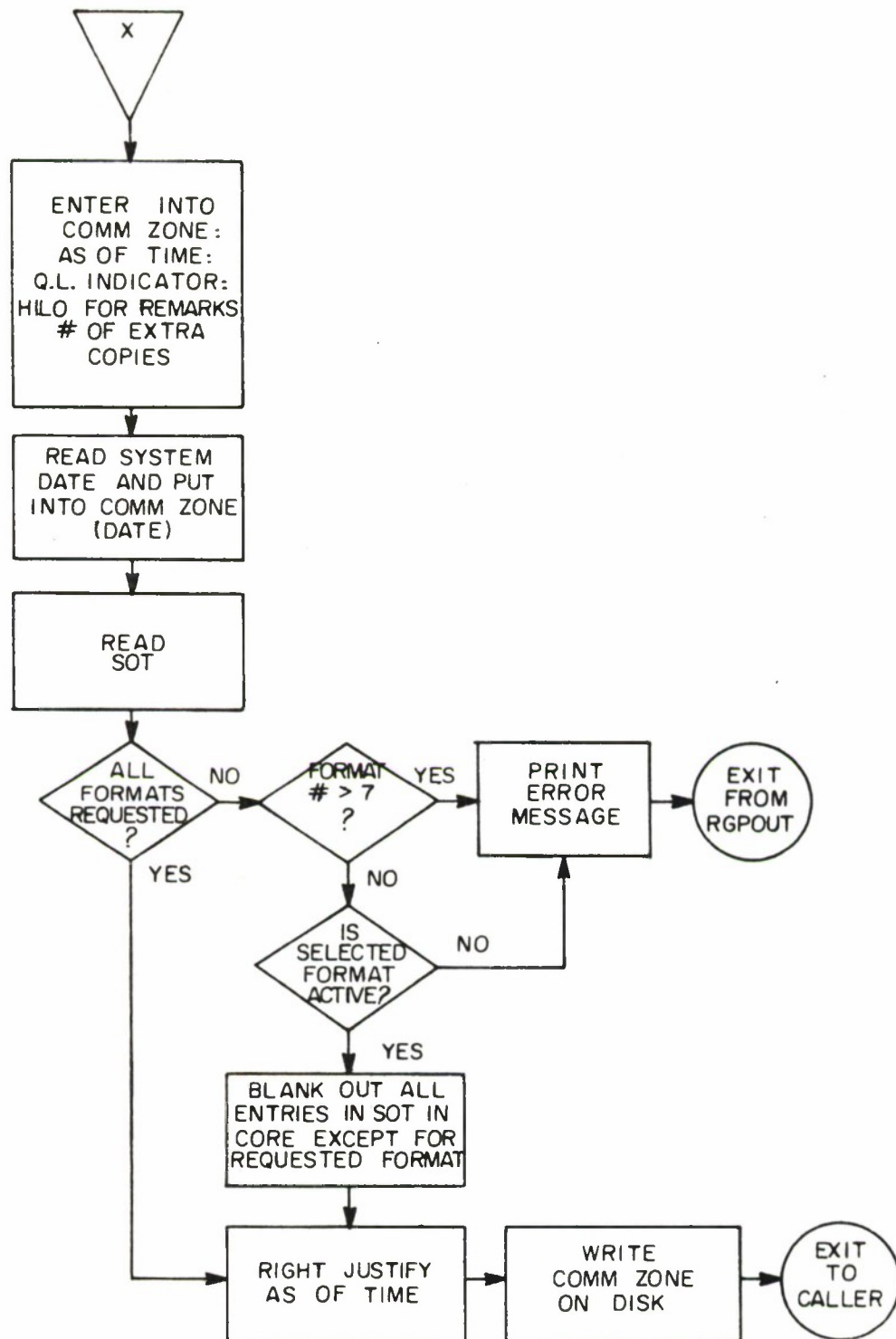
(Normal Return)

Description

ININT reads the I/O Communication Record from disk. The classification code is checked for U, C, S or T and causes the system attribute CLASS to be set to Unclassified, Confidential, Secret or Top Secret. Other data is taken from the I/O Communication Record and entered into the Communication Zone. The SOT Table is read into the Communication Zone. If only one specific format was requested, all other entries in the SOT Table are set to blanks. The Communication Zone is written on the disk from where it may be read back to re-initialize it between execution of output programs.

Return is made to RGPOUT.





UPSTAT

Purpose

UPSTAT is called by RGPOUT to complete the condensing of output statements which LOD could not completely condense. This only occurs in the case where the Serial File Tape is the product of a Query Language retrieval from a parallel file stored on system disk.

Calling Sequence

UPSTAT is called from RGPOUT by

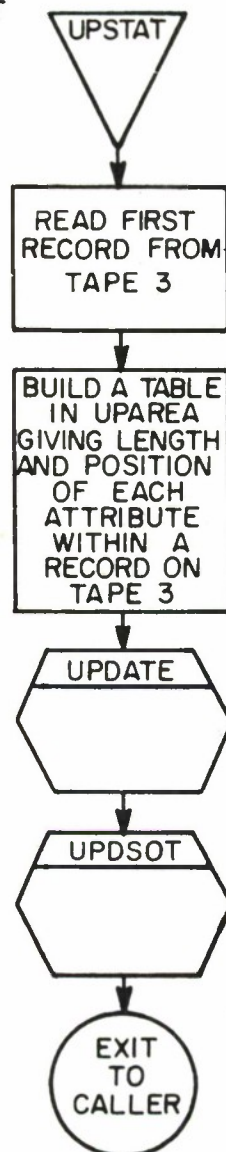
B UPSTAT
(Normal Return)

Description

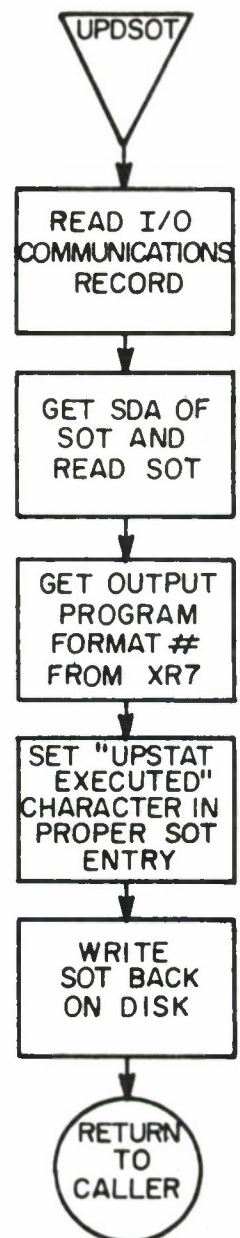
When LOD condenses the statements of an output program which uses a Query Language generated Serial File Tape as input, the statements cannot be completely condensed. In particular, LOD does not know the length or position in the Serial File Tape records of each attribute. LOD does know the order in which the attributes will appear on the Serial File Tape. In the case in point, LOD assigns each attribute a number corresponding to its relative position in the Serial File Tape records, and substitutes this number for each occurrence of an attribute name in an output statement.

When UPSTAT is executed, it reads the first record of the Serial File Tape and uses it to construct a table. The table so constructed contains the length and relative position on the Serial File Tape of each attribute. The condensed output program is read from disk, and the attribute numbers are now replaced with length and position data from the table. The fully condensed statements are written back onto disk, the "UPSTAT executed" indicator is set in the appropriate SOT entry, and return is made to RGPOUT.

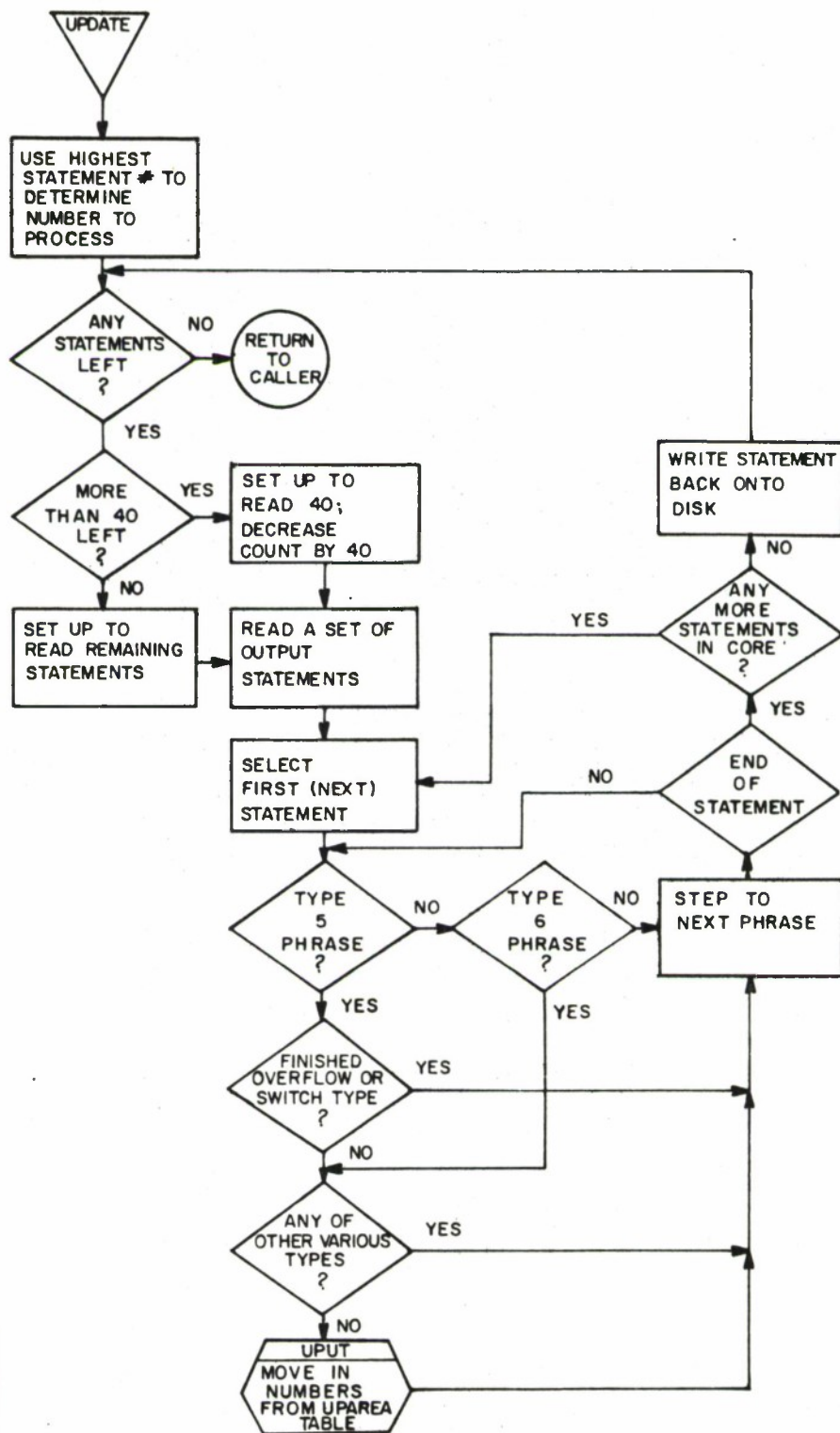
START



UPDSOT



IB - 26,205



118 - 26,267

PRINTO

Purpose

Print a line on the printer or read a record from the Serial File Tape.

Calling Sequence

PRINTO is entered from the REALO program. If a record is to be read from the Serial File Tape, the call is

B INPUT
(Normal return)

If the output image is to be printed, the call is

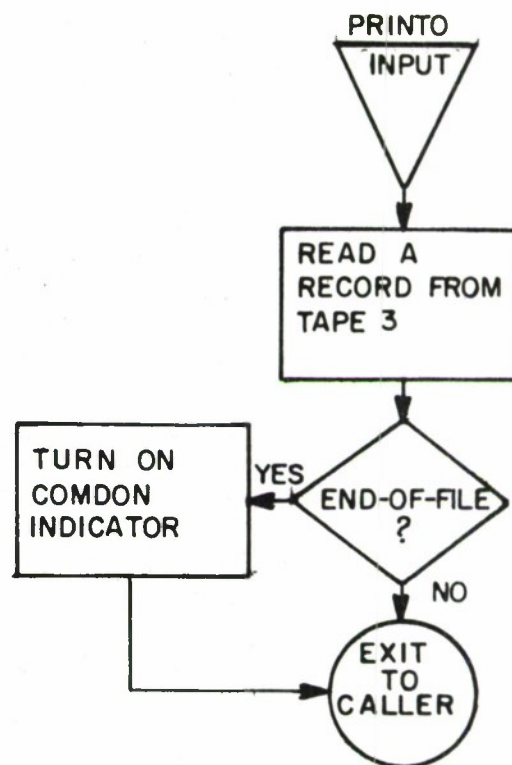
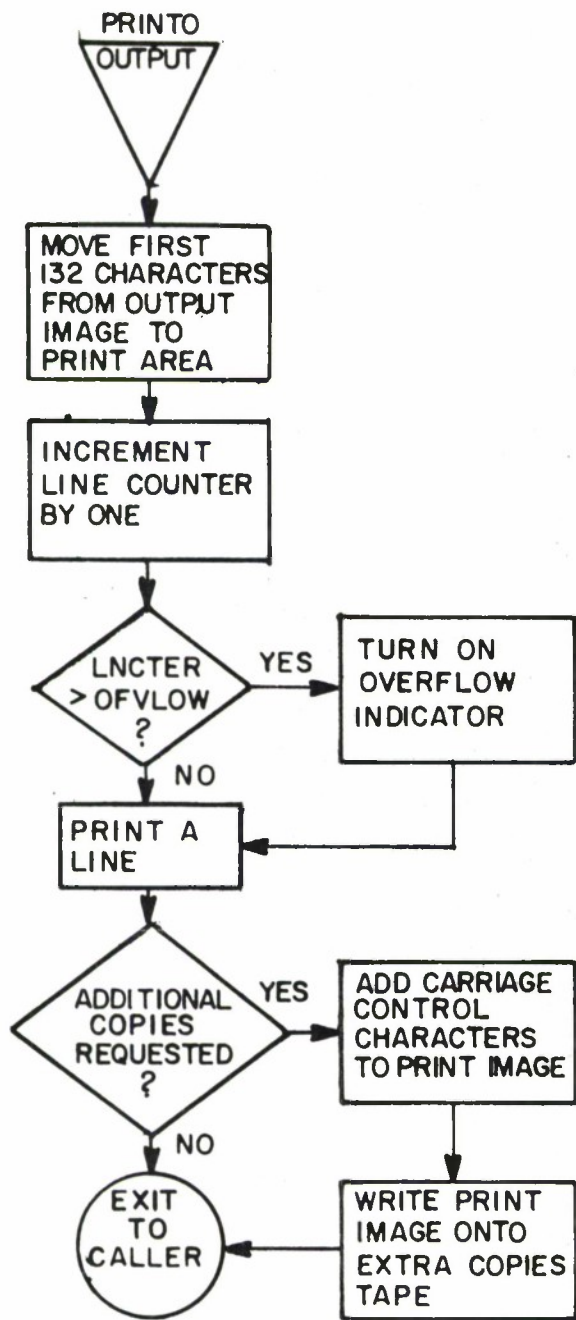
B OUTPUT
(Normal return)

Description

If entry is made at INPUT, one record is read from the Serial File Tape. If an end-of-file is encountered, the indicator COMDON is turned on in the Communications Zone.

If entry is made at OUTPUT, the first 132 characters of the output image are printed. The line counter is incremented, and if it exceeds the value in OVFLOW, the overflow indicator is turned on.

If additional copies of the printed report were requested, the print image is also written onto the extra copies tape. Three characters are added to the record on tape to indicate any blank lines or page ejects which should be accomplished before this line is printed. (This spacing information will be interpreted by the PRTTAP program which generates additional copies of the printed report from the extra copies tape.)



CARDO

Purpose

Punch a card from the output image or read a record from the Serial File Tape.

Calling Sequence

CARDO is entered from the REALO program. If a record is to be read from the Serial File Tape, the call is

B INPUT

(normal return)

If a card is to be punched, the call is

B OUTPUT

(normal return)

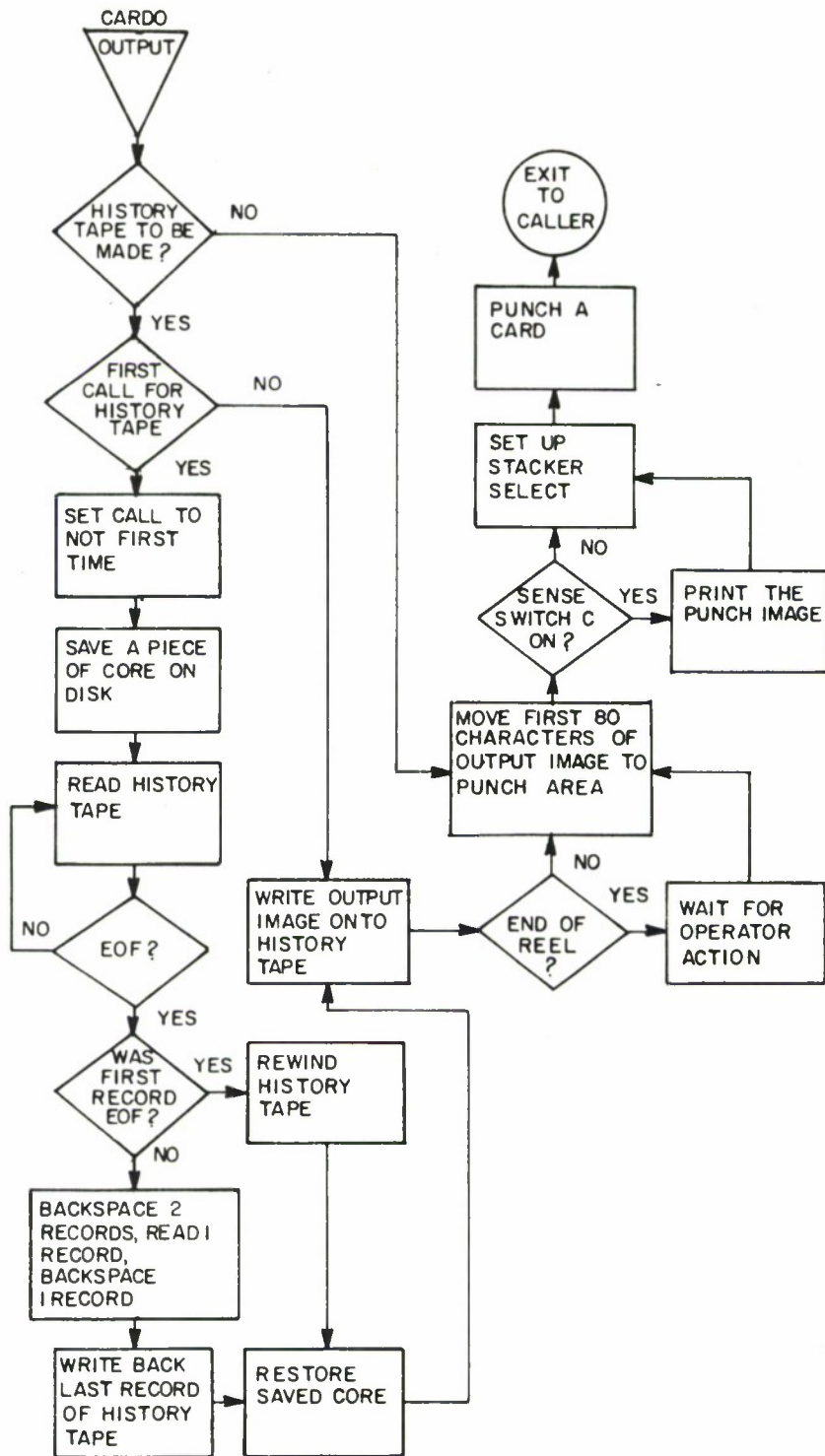
Description

If entry is made at INPUT, one record is read from the Serial File Tape. If an end-of-file is encountered, the indicator COMDON is turned on in the Communications Zone.

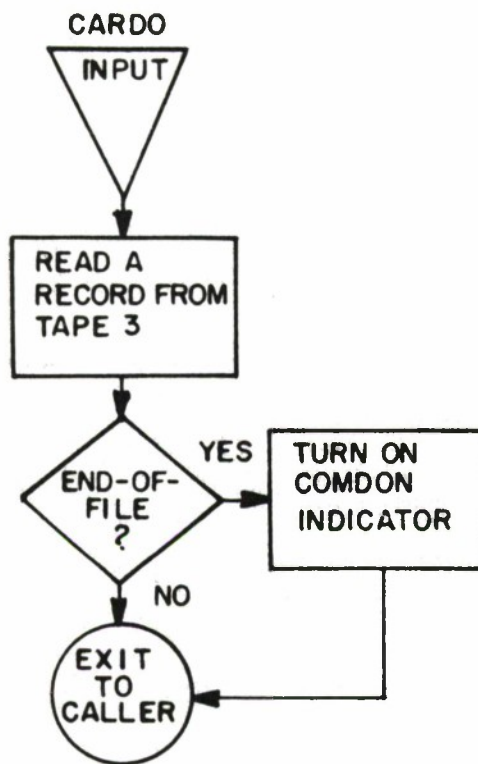
If entry is made at OUTPUT, a check is made to see if a history tape was requested. If a history tape was requested, the first entry at OUTPUT causes the history tape to be spaced until an end-of-file is detected, and then positioned so that records may be added to the file. The output image is then written on the history tape.

If Sense Switch C is on, the output image is printed.

Next the stacker select character is acquired from the Communication Zone, and the first 80 characters of the output image are punched in a card.



IB-26,269



TAPEO

Purpose

Write an output image onto magnetic tape, write a Header record on magnetic tape, write a Trailer record on magnetic tape or read a record from the Serial File Tape.

Calling Sequence

TAPEO is entered from the REALO program. If a record is to be read from the Serial File Tape, the call is

B INPUT
 (normal return)

If the output image is to be written on tape, the call is

B OUTPUT
 (normal return)

If a Header is to be written on tape, the call is

B HEADER
DCW 'NNN'

where NNN is the number of characters of the output image to be used as a Header record. If a Trailer record is to be written on tape, the call is

B TRAILER
DCW 'NNN'

where NNN is the number of characters of the output image to be used as a Trailer record.

Description

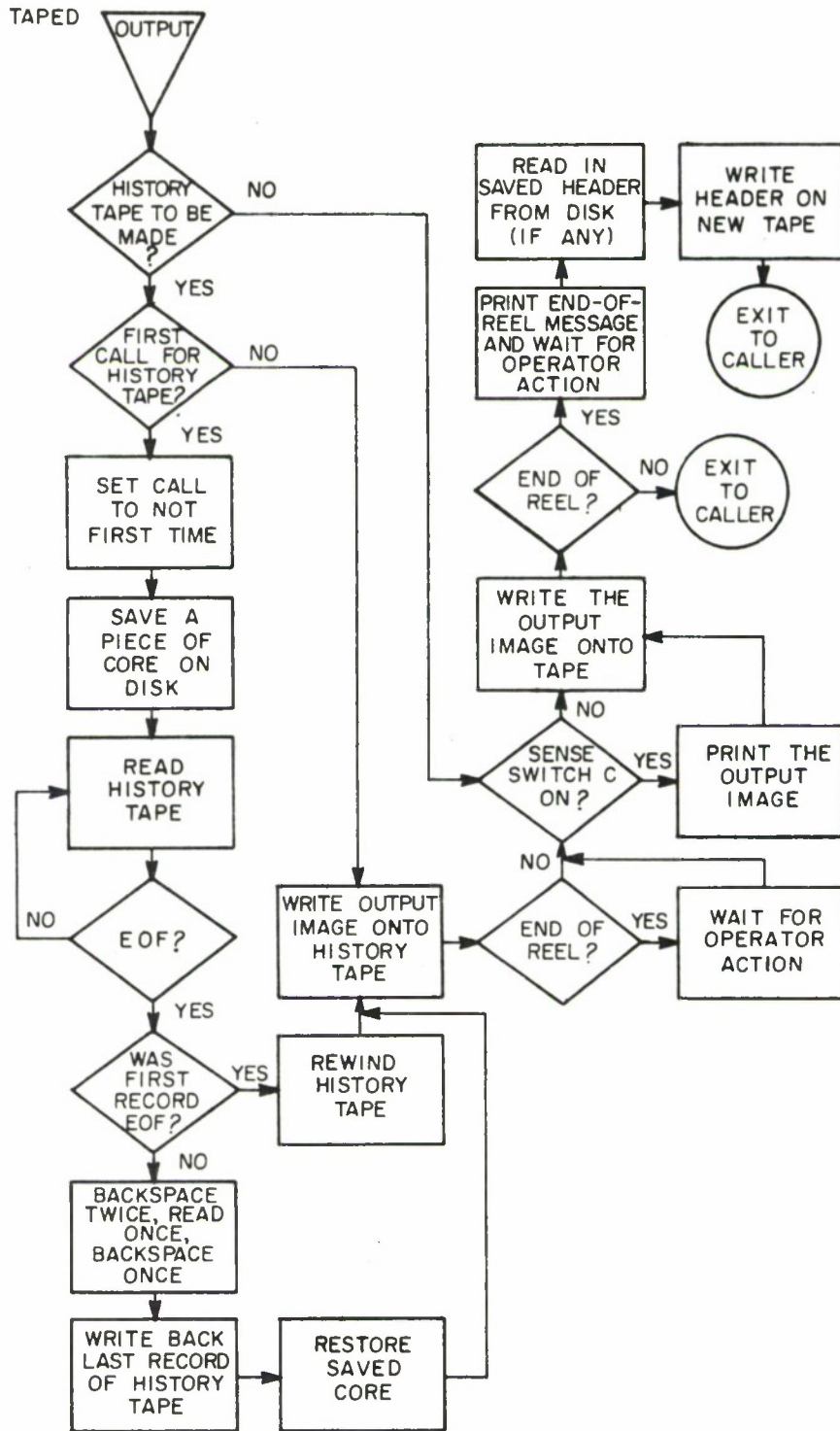
If entry is made at INPUT, one record is read from the Serial File Tape. If an end-of-file is encountered, the indicator COMDON is turned on in the Communication Zone.

If entry is made at OUTPUT, a check is made to see if a history tape was requested. If a history tape was requested, the first entry at OUTPUT causes the history tape to be spaced until an end-of-file is detected, and then positioned so that records may be added to the file. The output image (up to 1000 characters) is

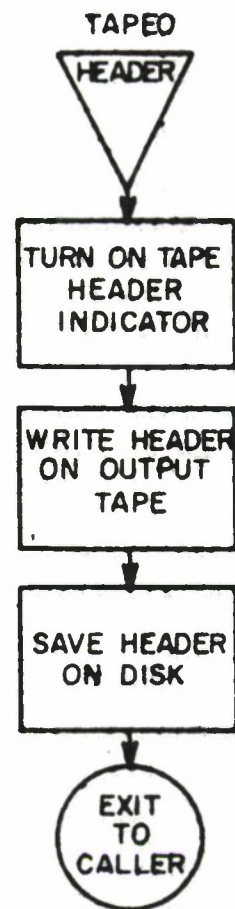
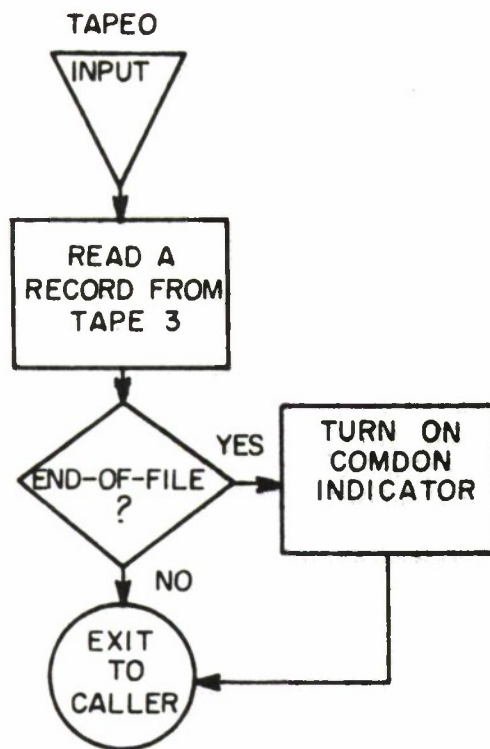
then written on the history tape. If Sense Switch C is on, the first 132 characters of the output image are printed. The output image is then written on the output tape.

If entry is made at **HEADER**, the Header indicator is turned on, the first NNN characters of the output image are written on the output tape, and are also saved on the system disk (if a subsequent entry at **OUTPUT** results in an end-of-reel condition on the output tape, the saved Header will be written as the first record of the next reel).

If entry is made at **TRAILER**, the first NNN characters of the output image are written on the output tape.



18-26,188



REALO

Purpose

Interpretively execute an output program to produce the desired report.

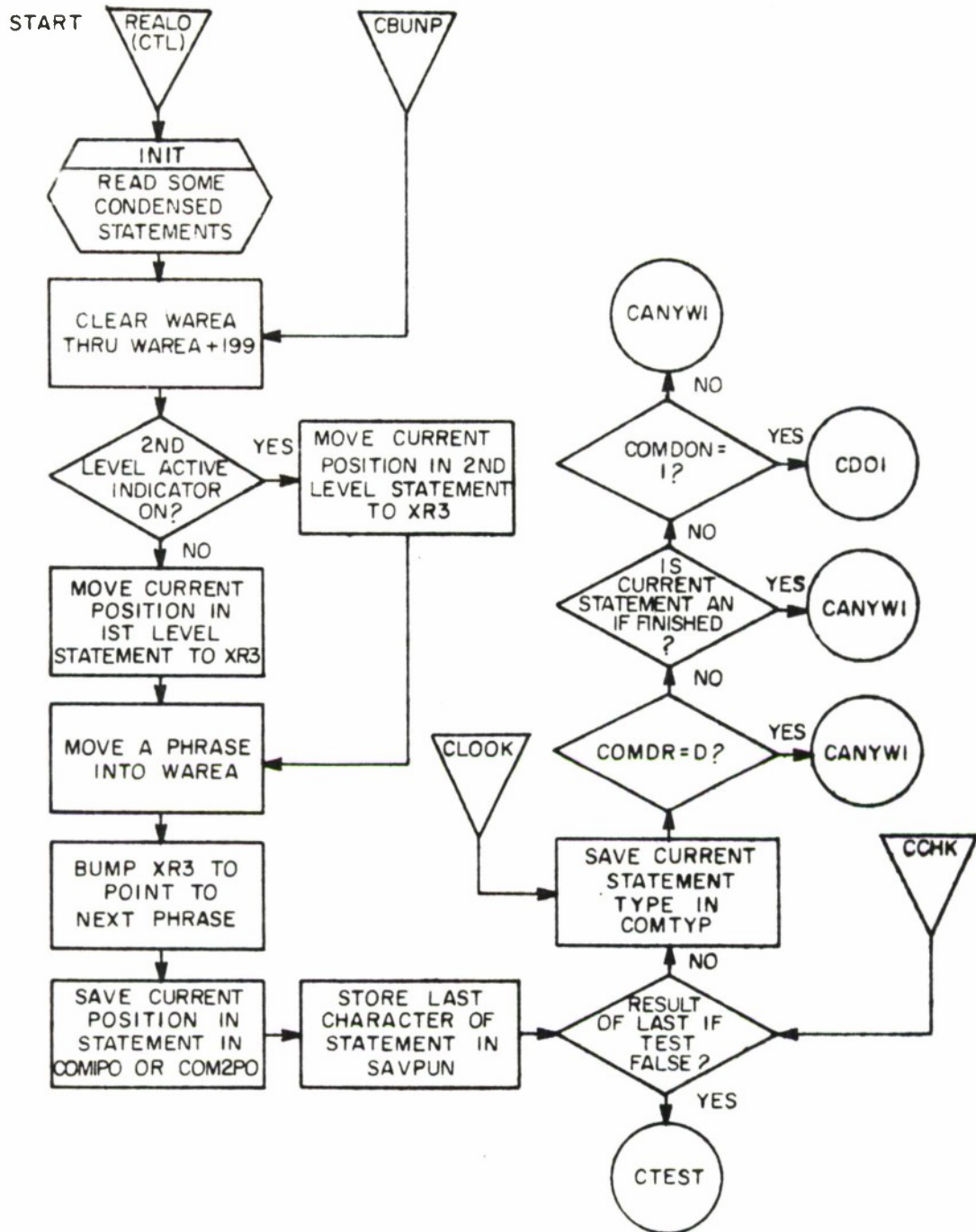
Calling Sequence

REALO is called internally from RGPOUT by

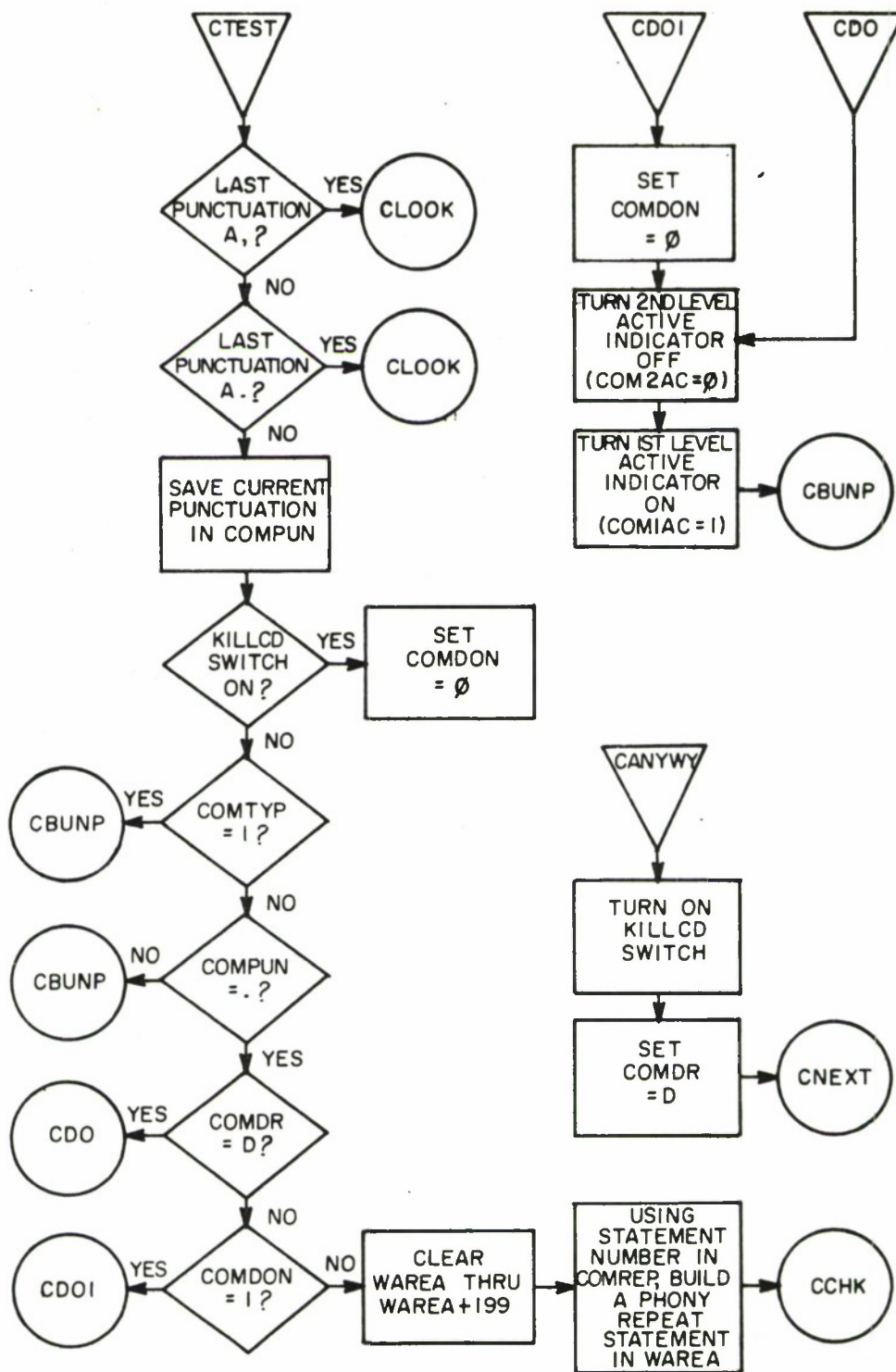
B CTL
(normal return)

Description

An initialize routine is first executed to read into core as much of the condensed output program as will fit. Statements are kept in core just as they appeared on disk, in 245 character records. Each statement, as it is required, is moved to a buffer at WAREA from where it is actually executed. The current statement is examined one phrase at a time. Each phrase begins with a number from 1 to 6 indicating the phrase type. Recognition of the type causes a transfer to the appropriate TYPE routine, where the phrase is completely identified and executed. Section 6.4.4 under LOD shows the phrases grouped by type, and the format of the condensed phrases. Execution of an END phrase causes control to be passed from REALO back to RGPOUT.

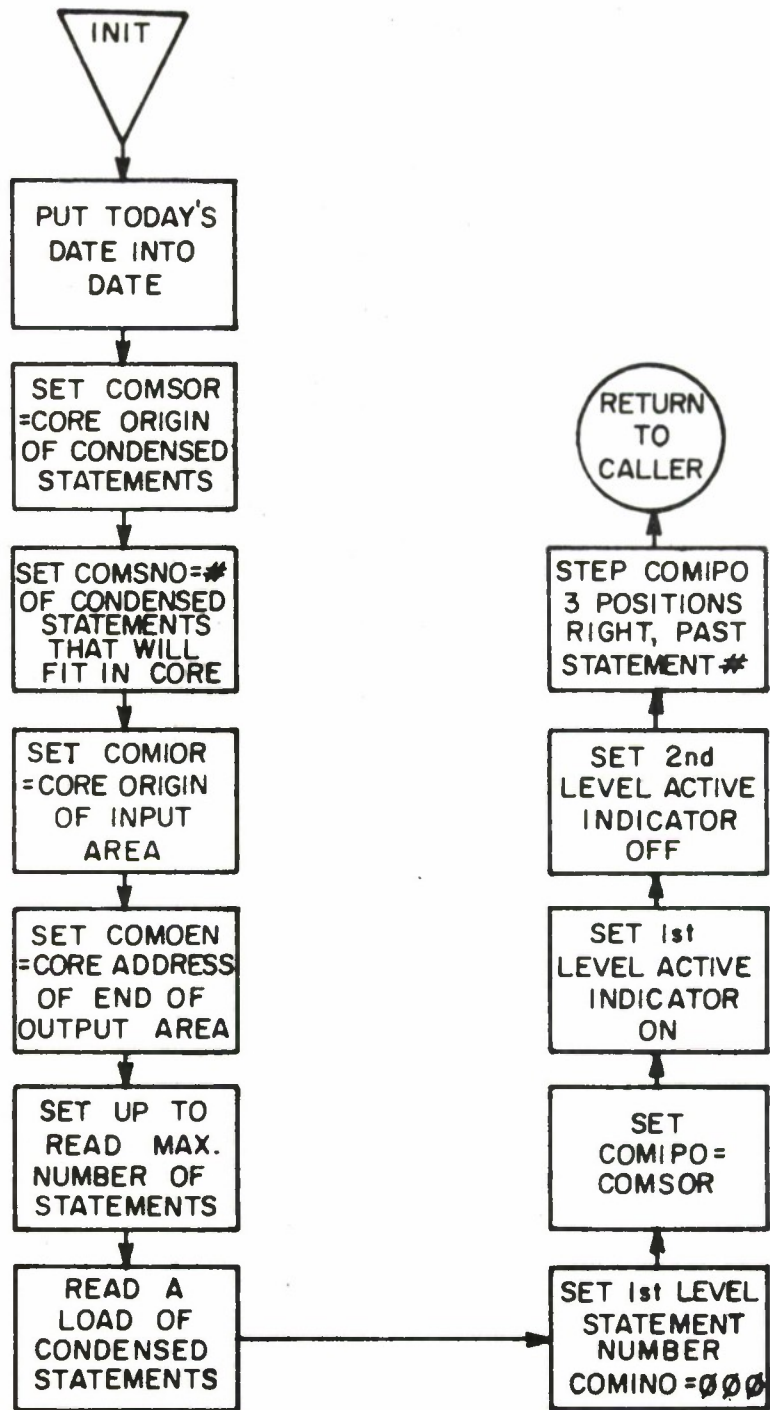


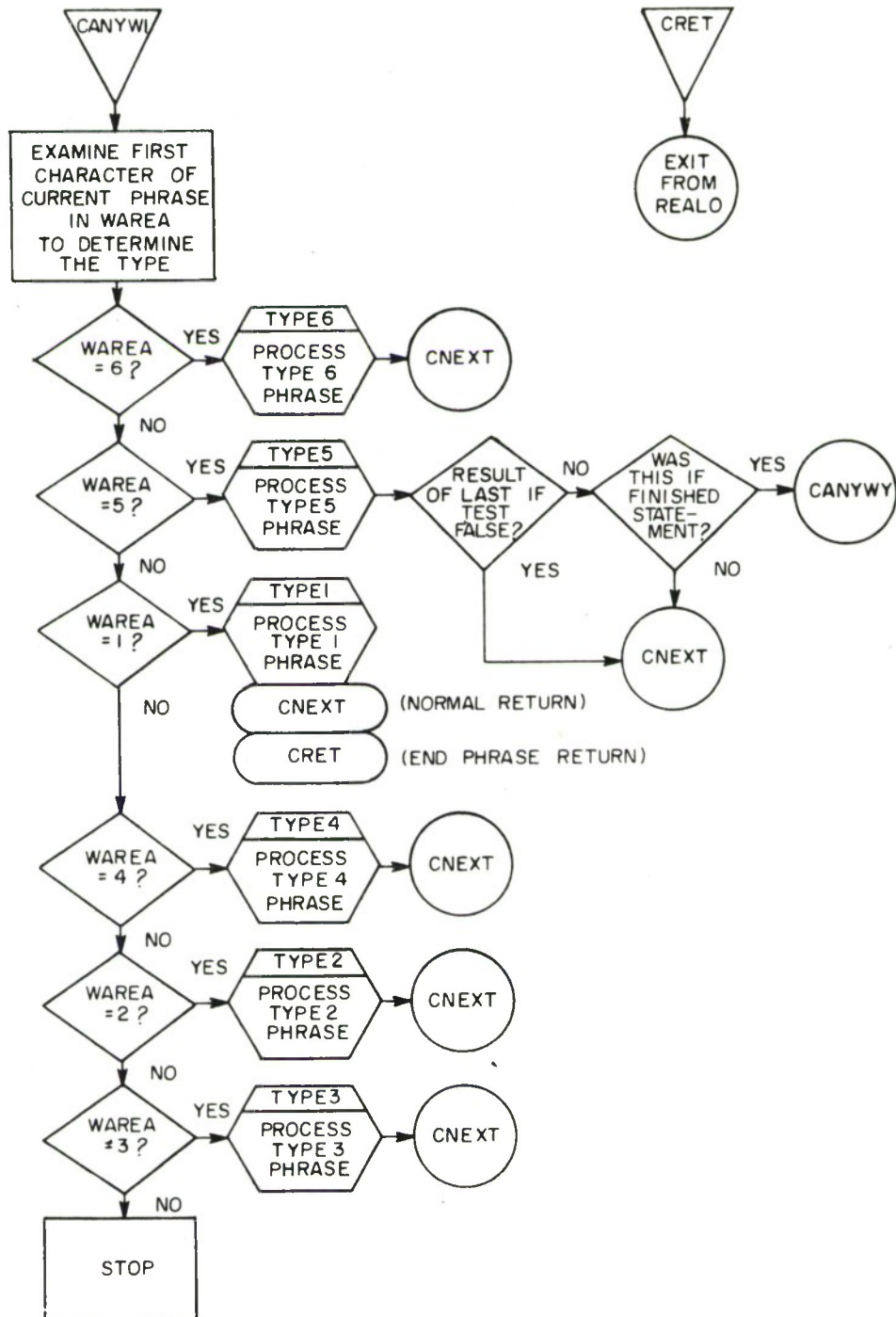
18-26,212



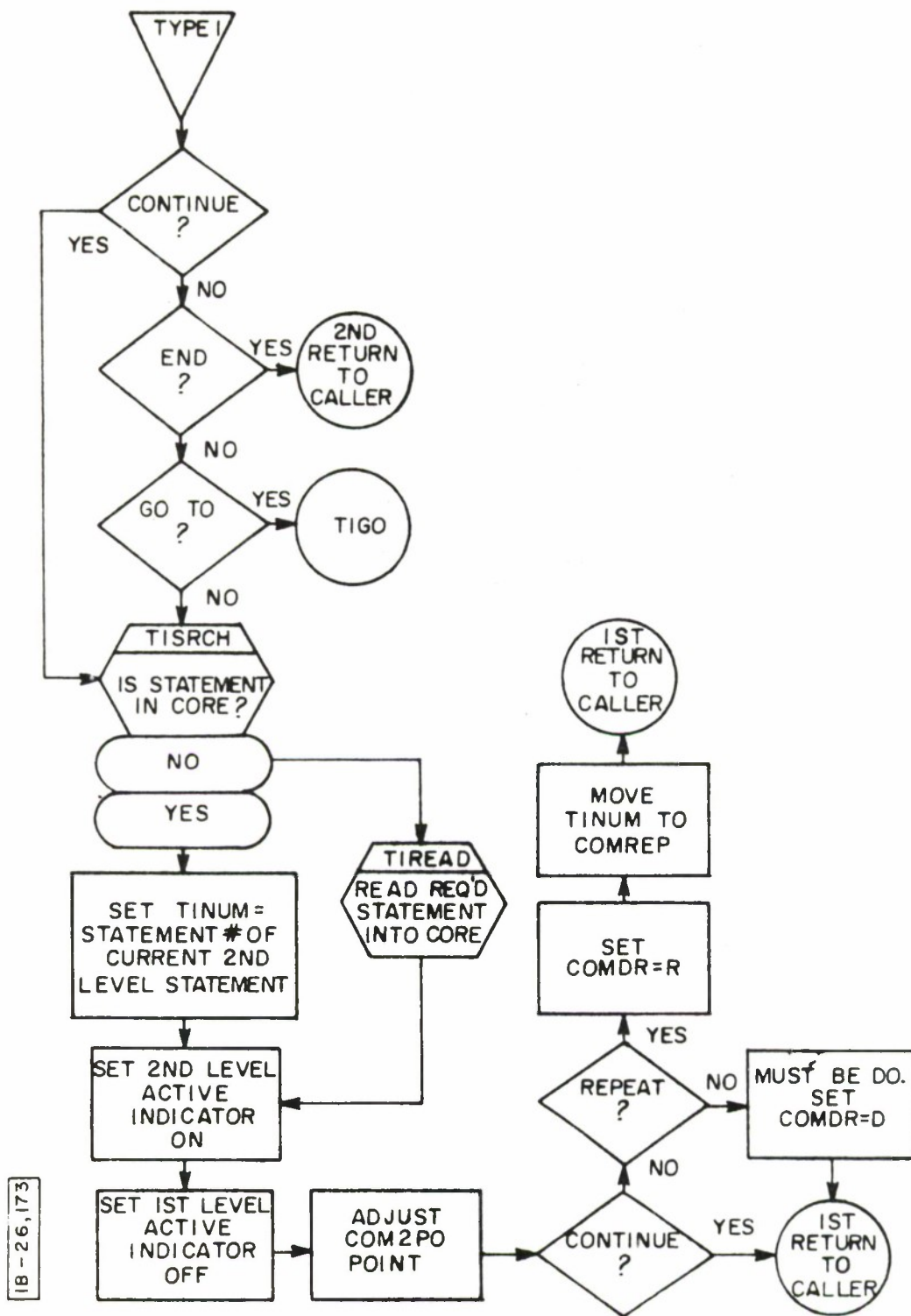
11B-26,202

IB-26,201

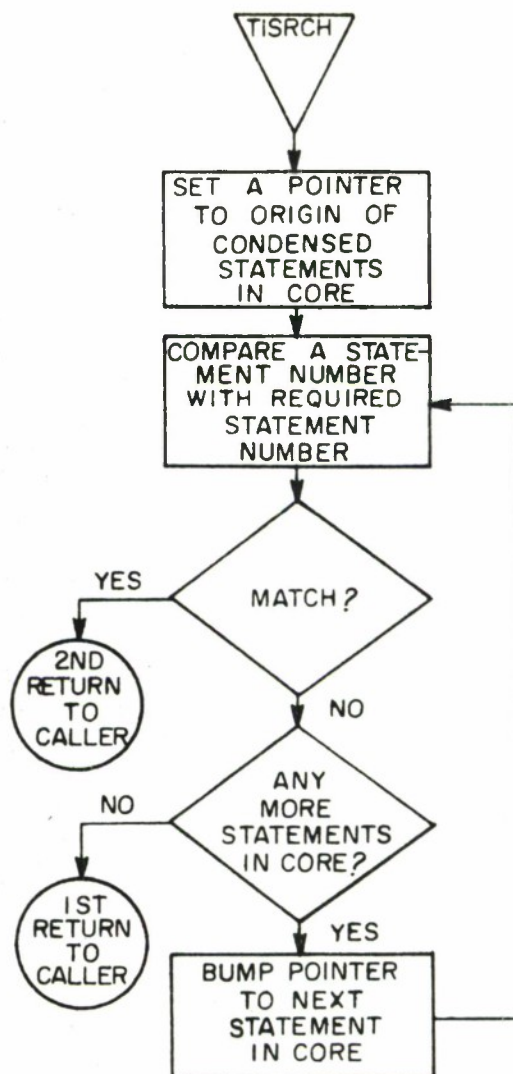
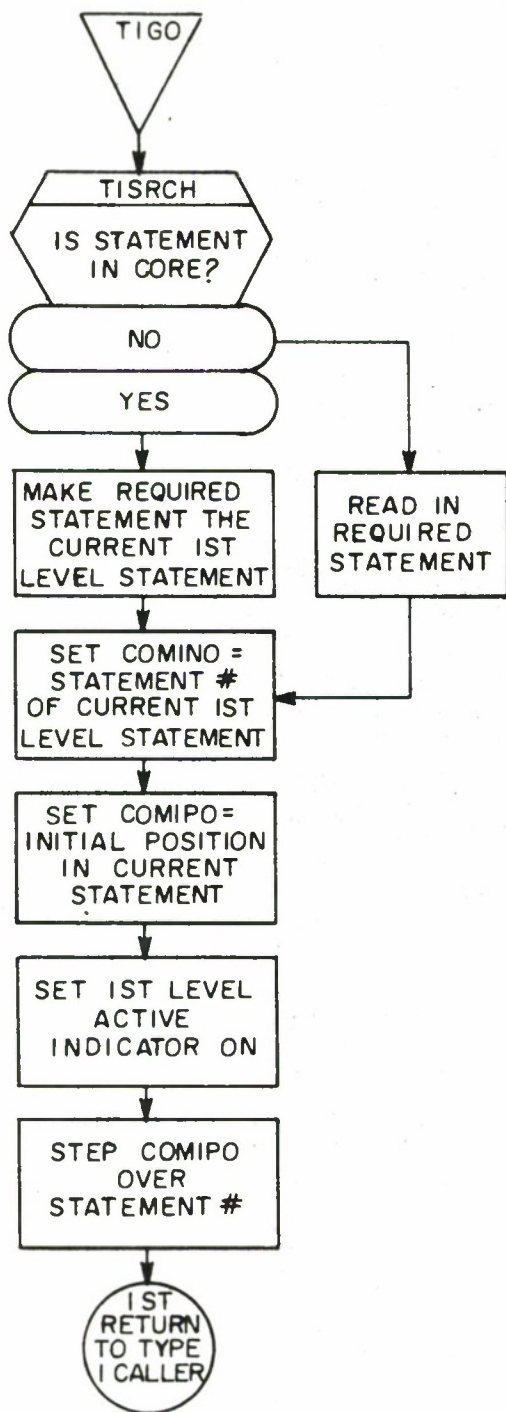


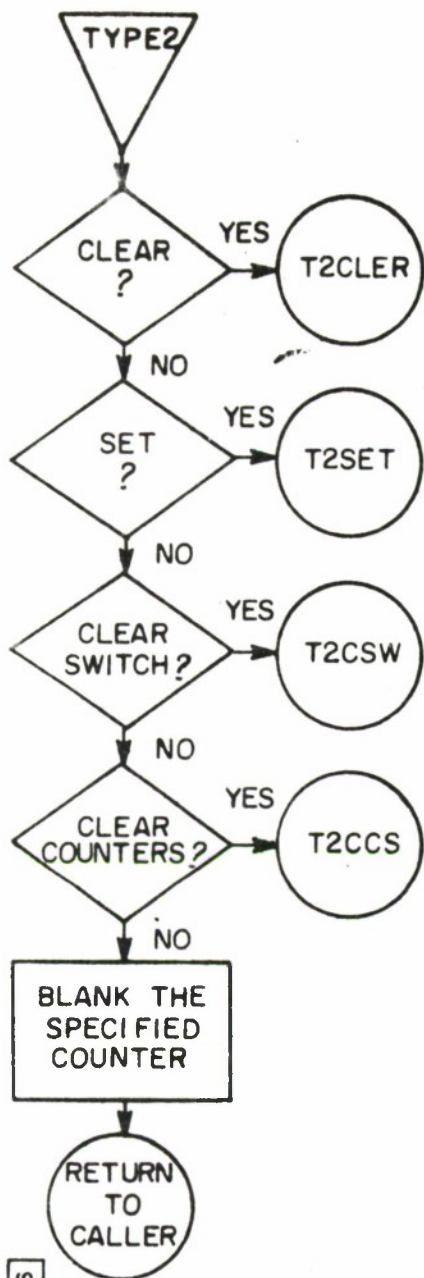


18-26,172

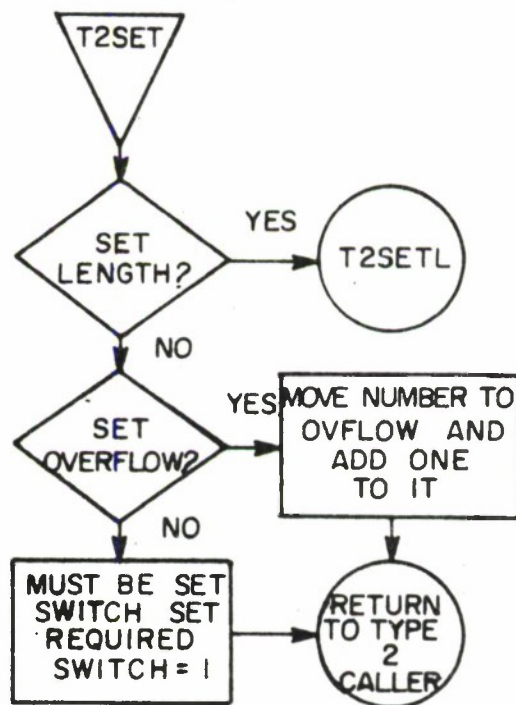
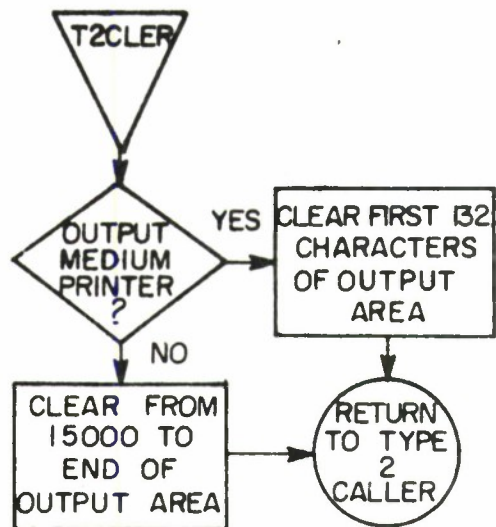


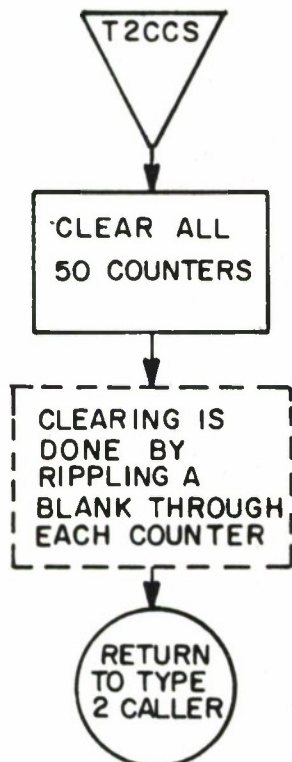
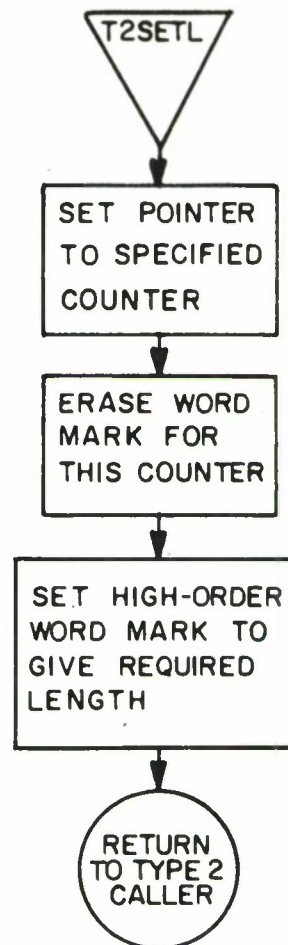
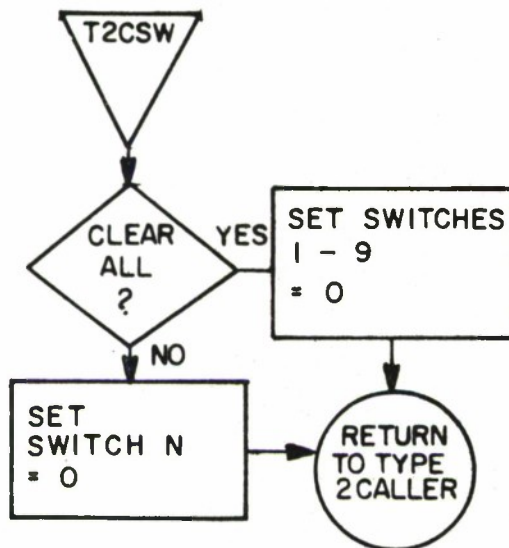
IB - 26, 173



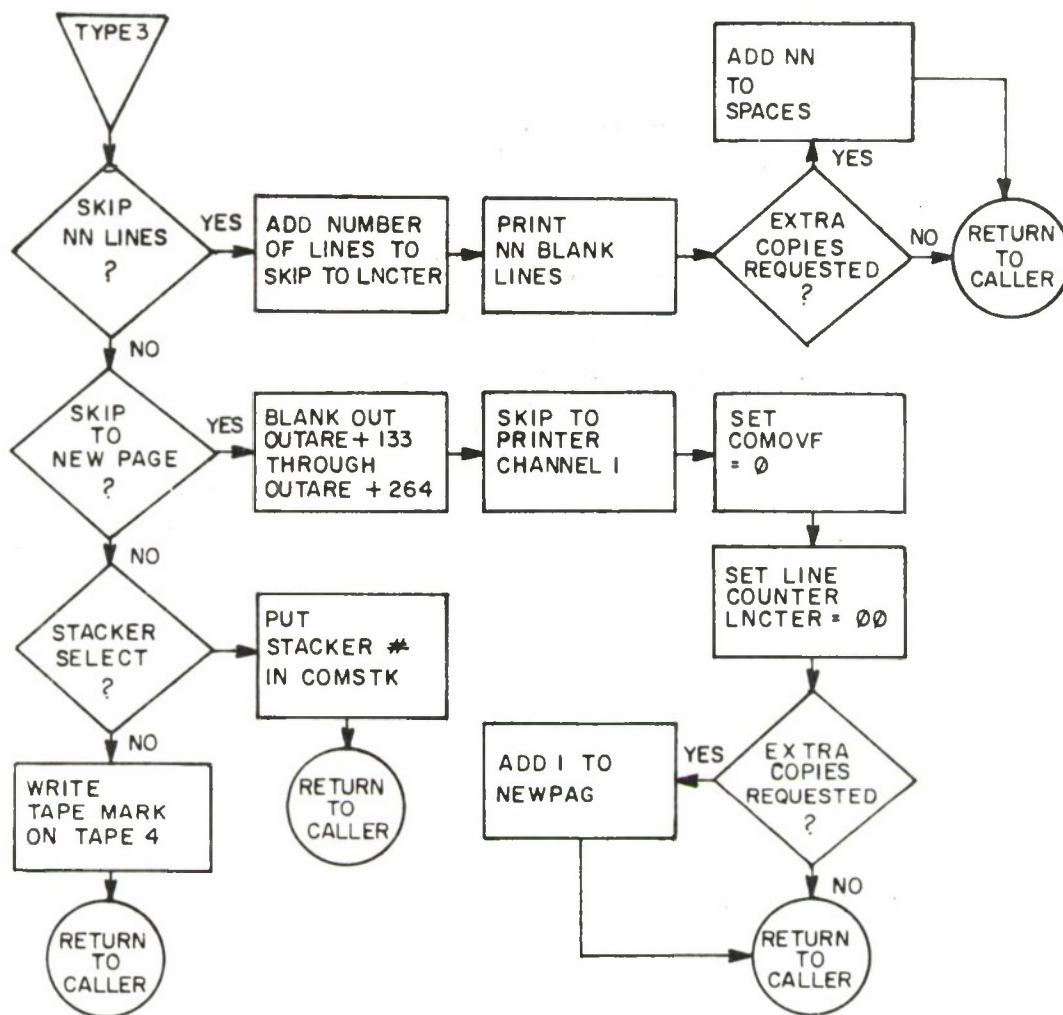


IB - 26,226

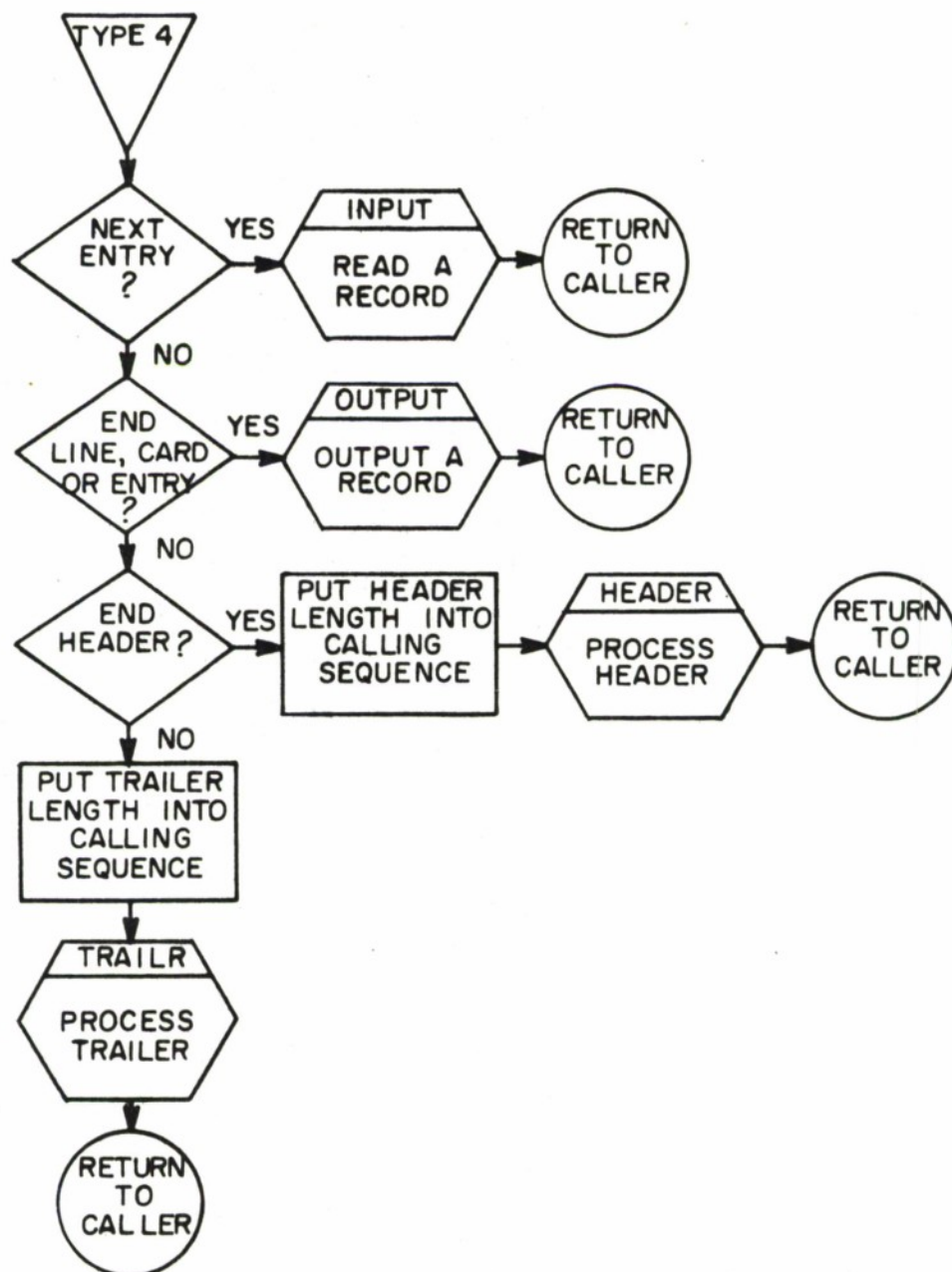




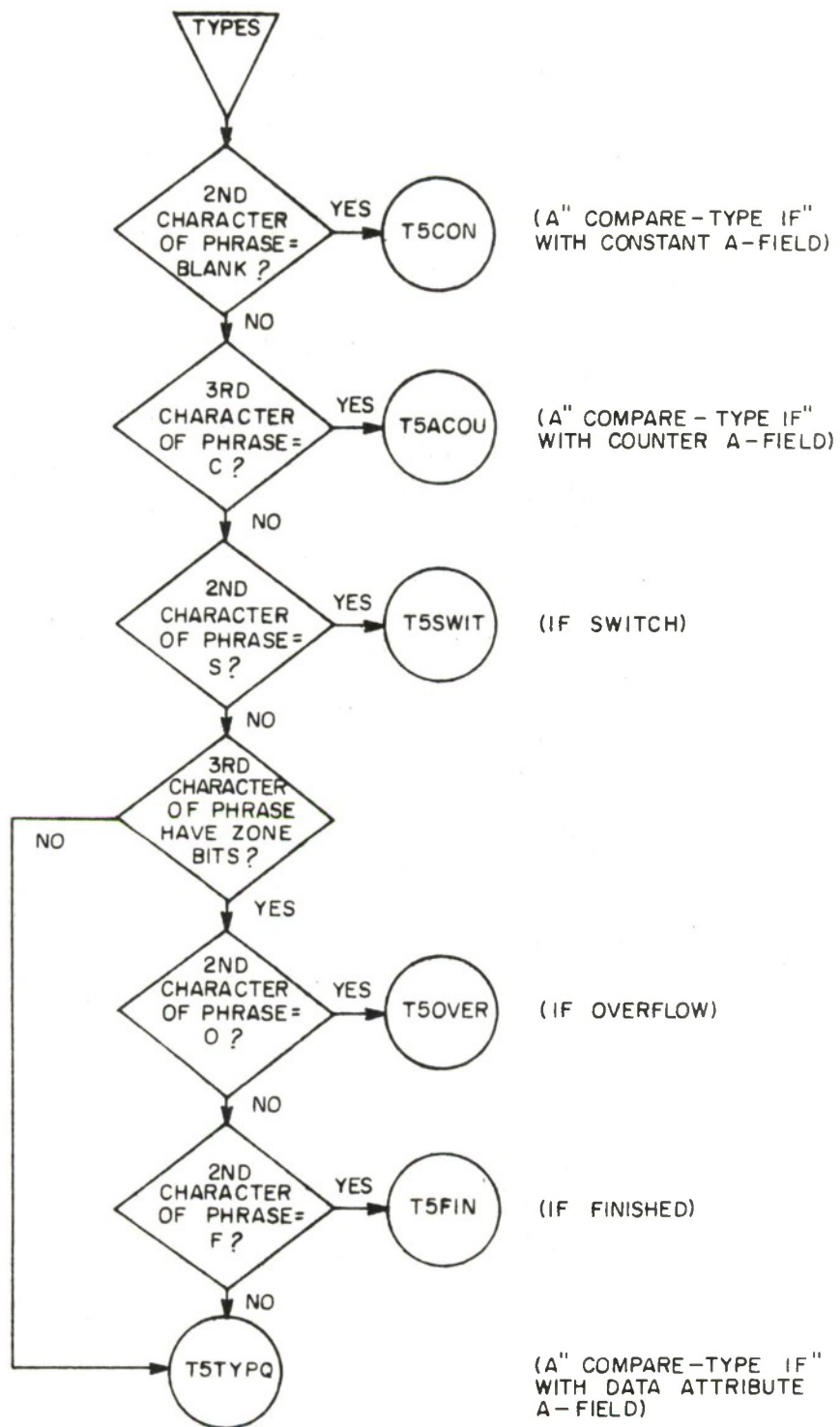
18-26,271



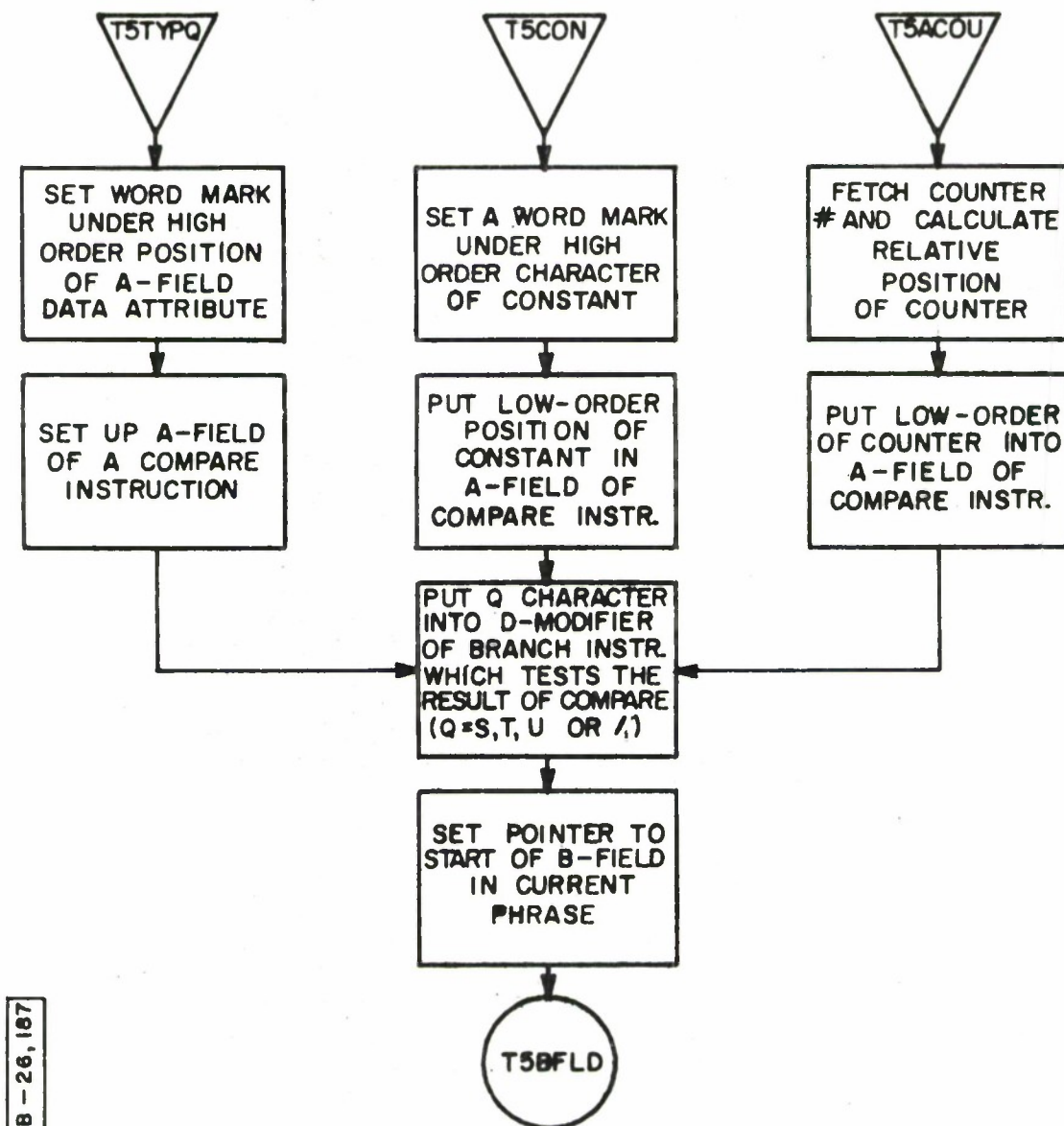
IB-26,272

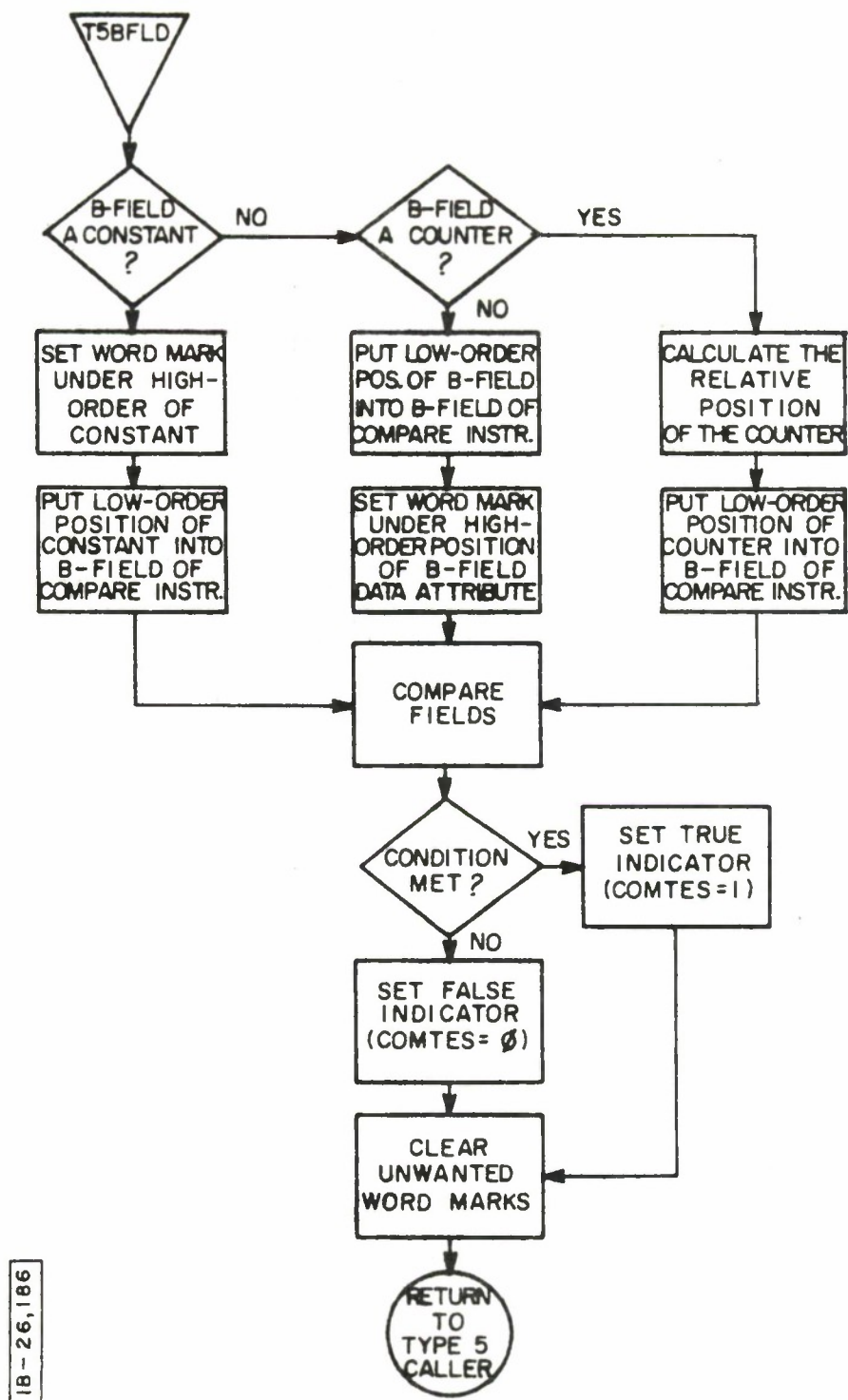


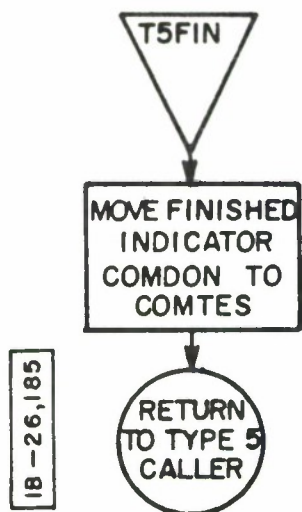
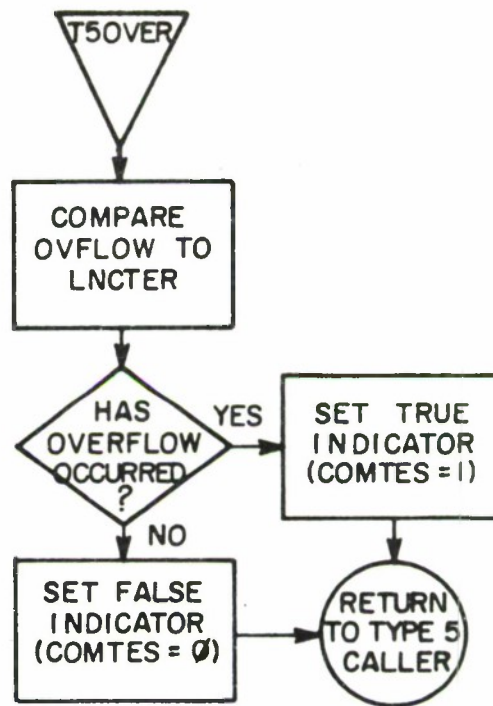
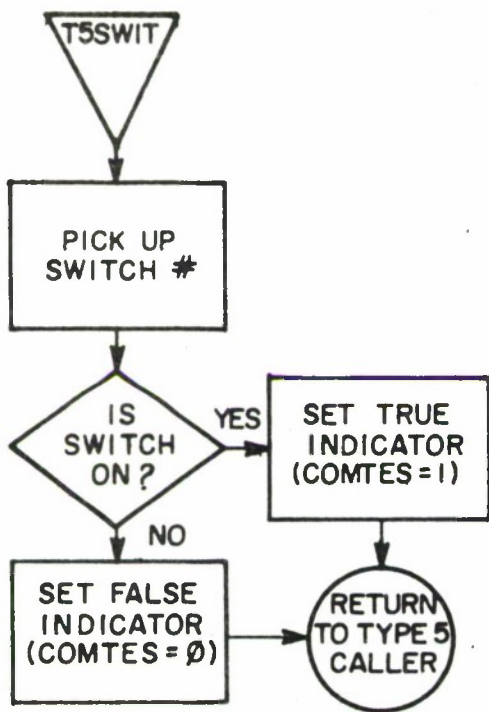
IB-26,218



18-26,217

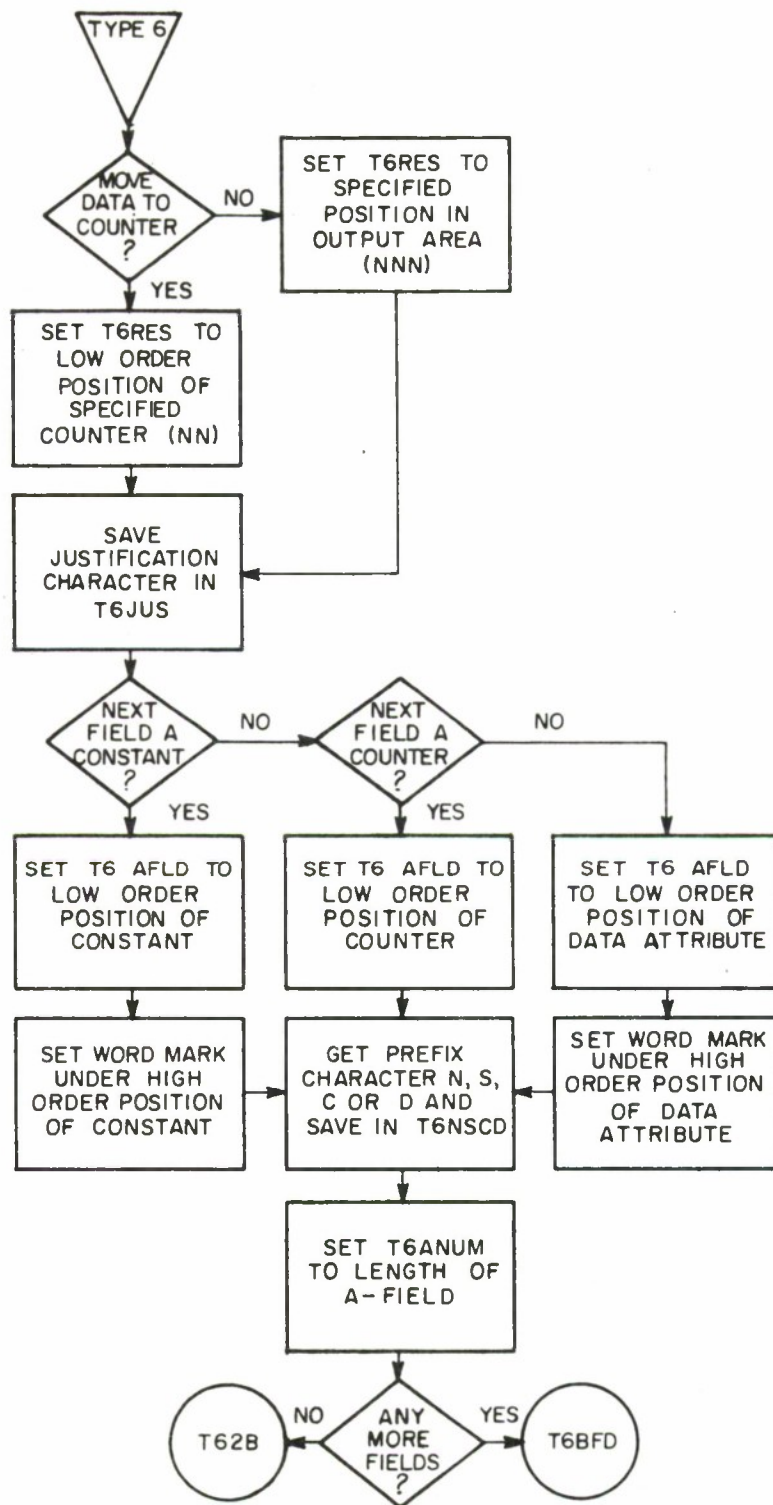


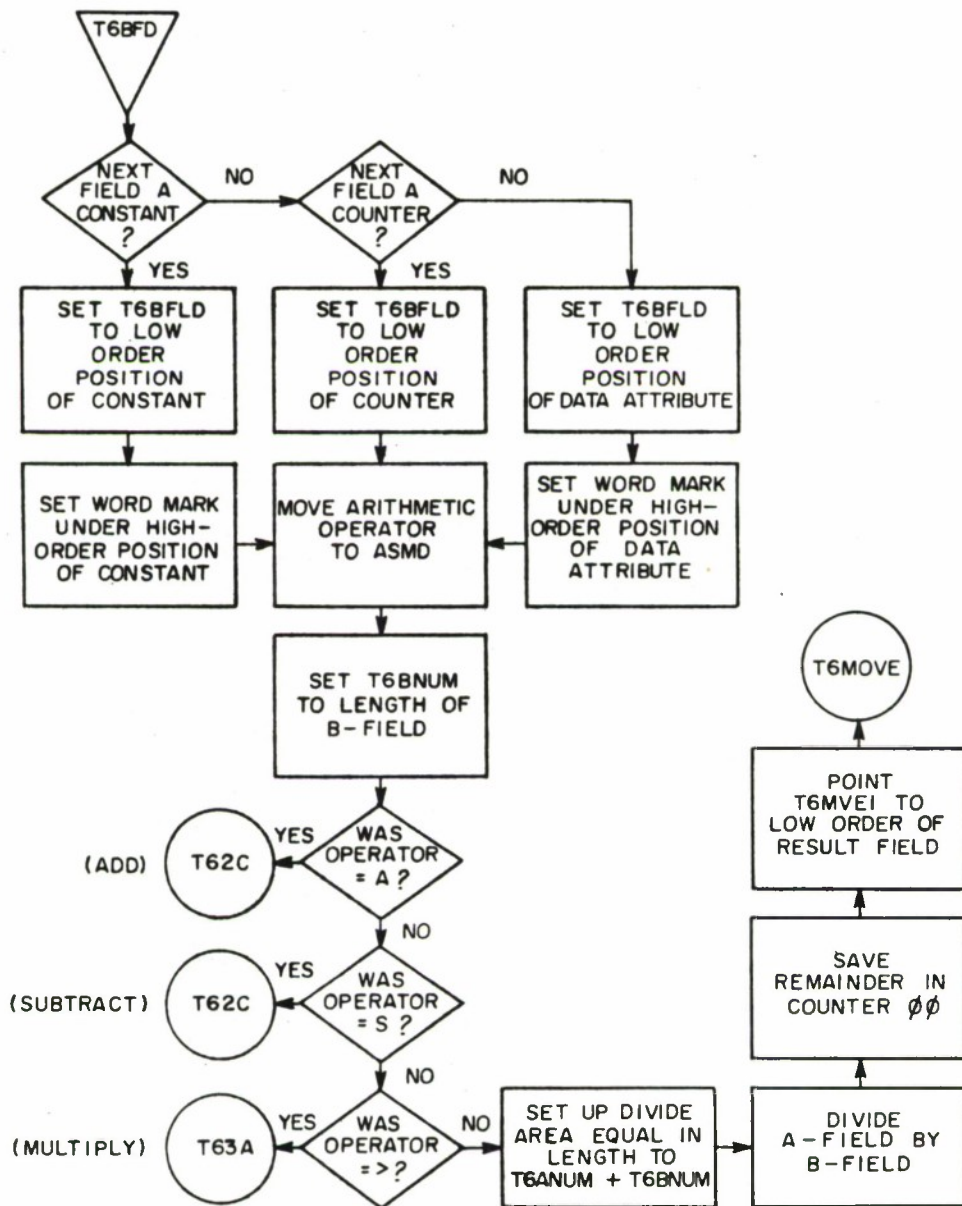




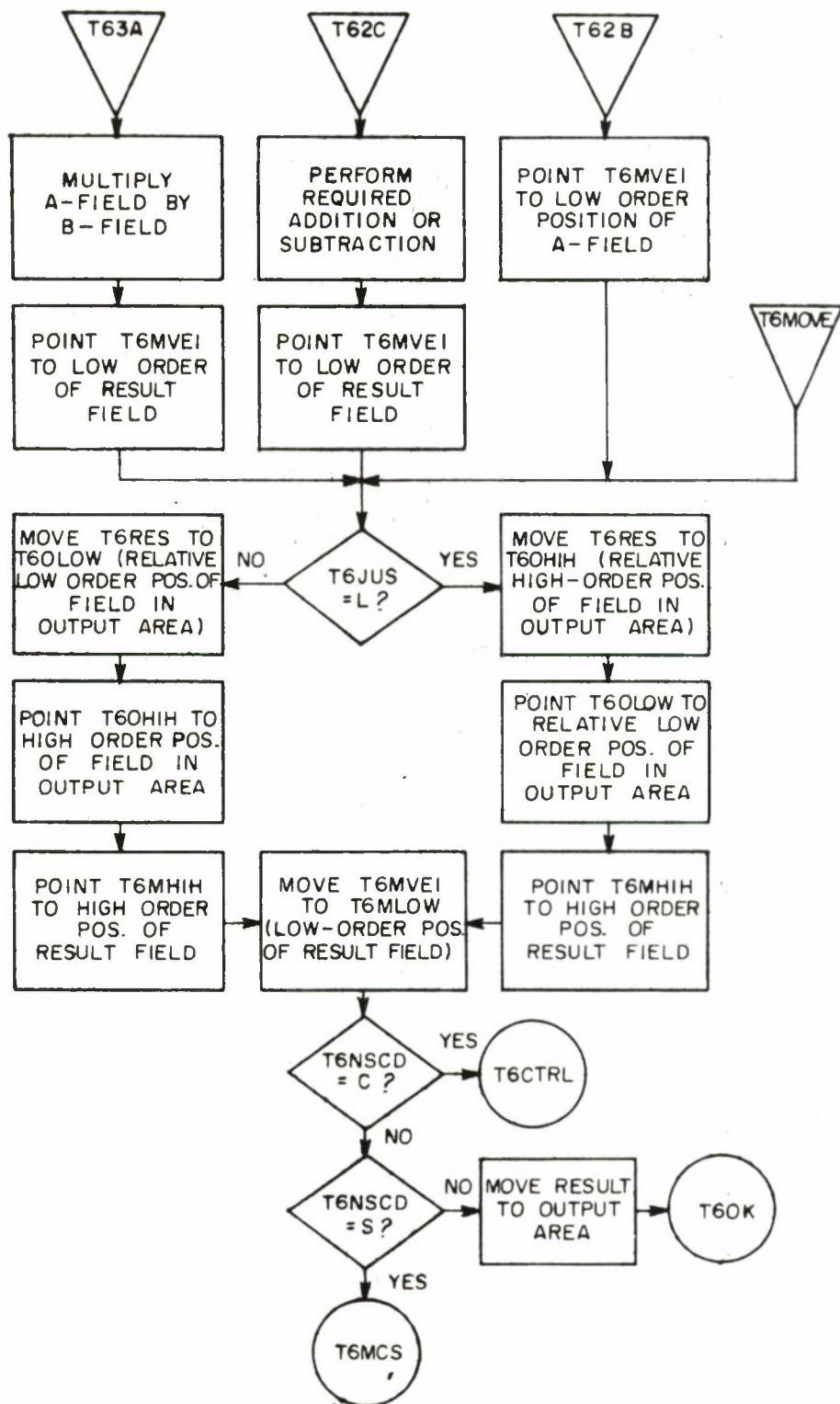
1B-26,185

IB-26,184

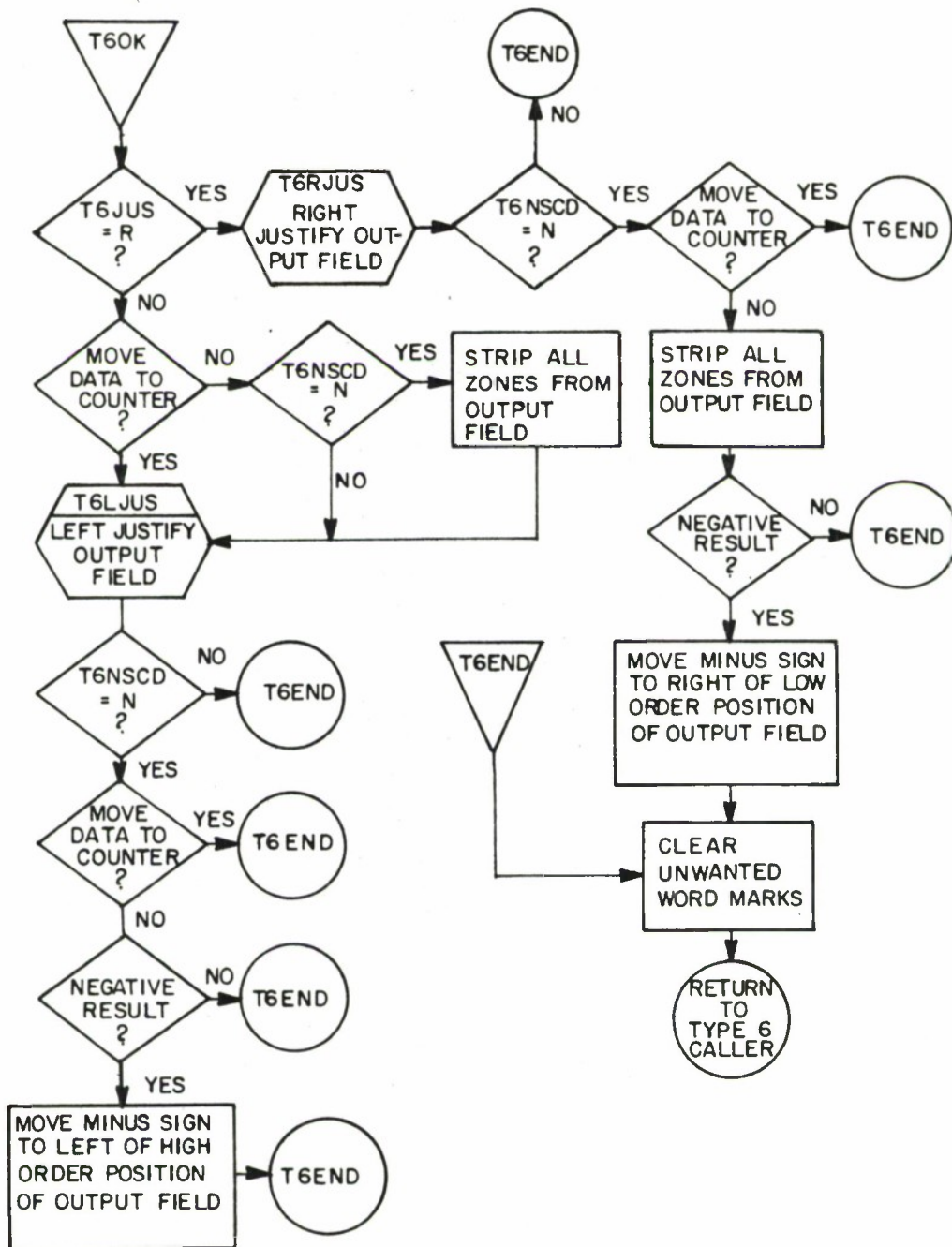




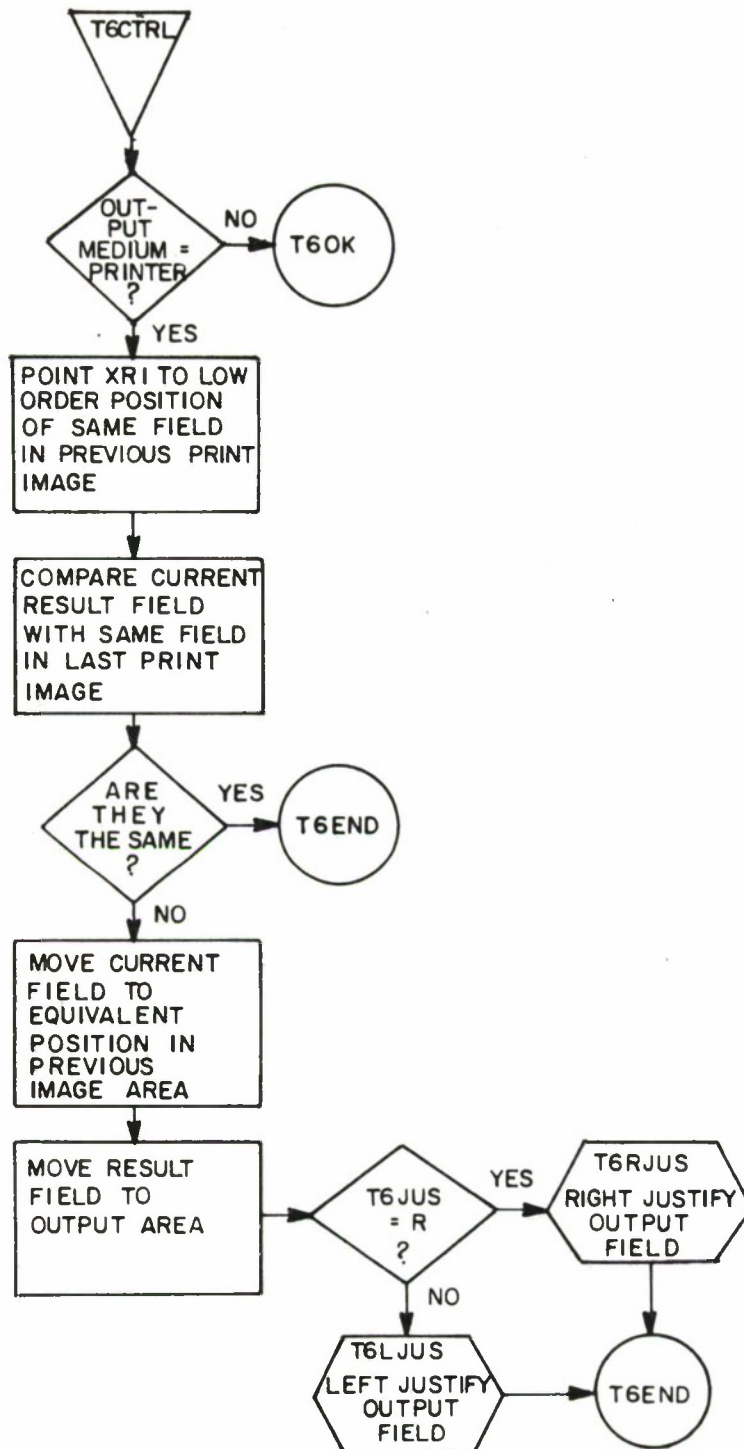
1B-26,232



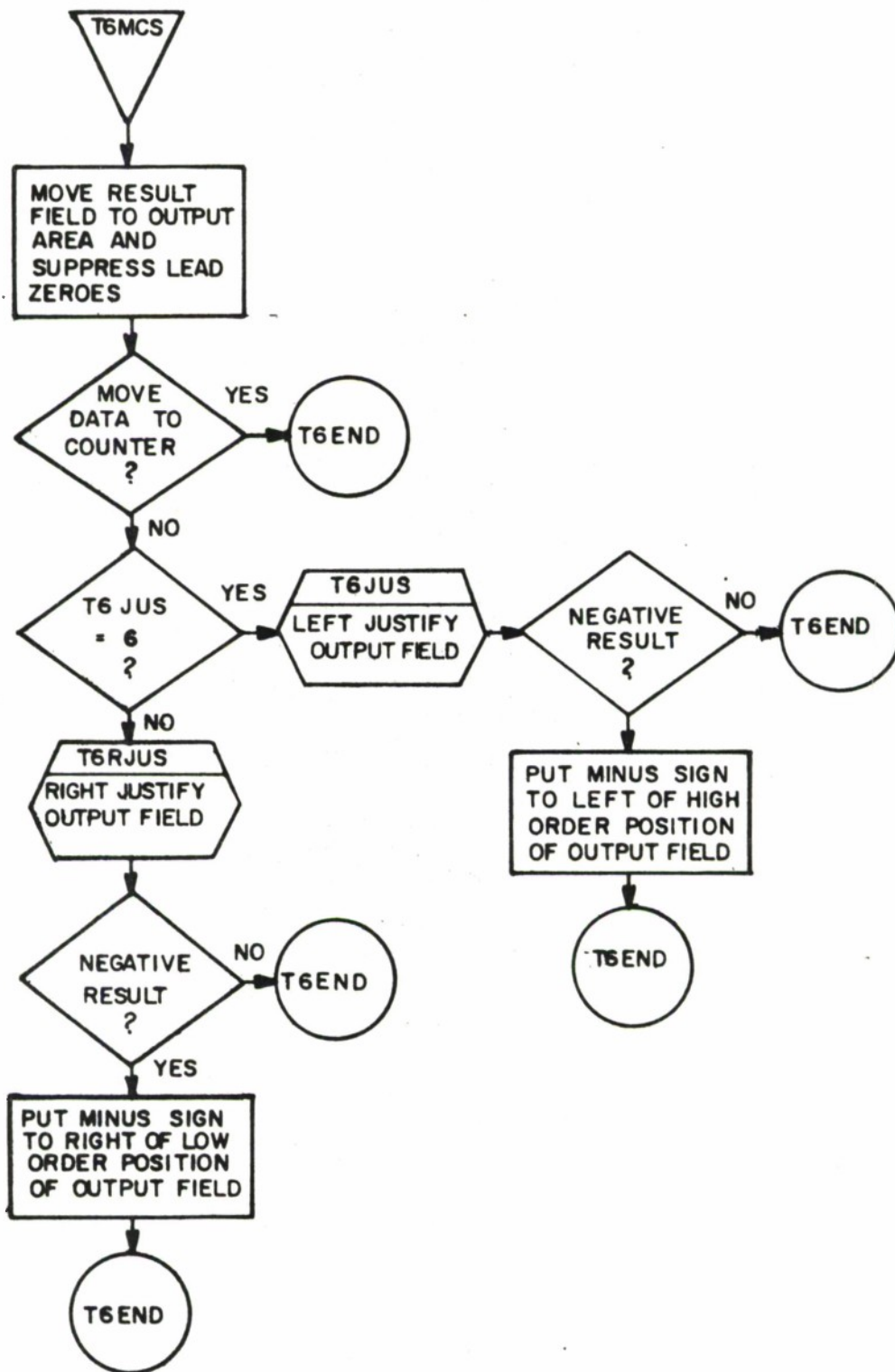
18-26,231



1B-26,273



IB - 26,274



IB-26,275

REMARK

Purpose

Print or punch remarks which were stored on disk during the input phase of a REPGEN run.

Calling Sequence

REMARK is loaded and called from RGPOUT by

B REMARK
 (normal return)

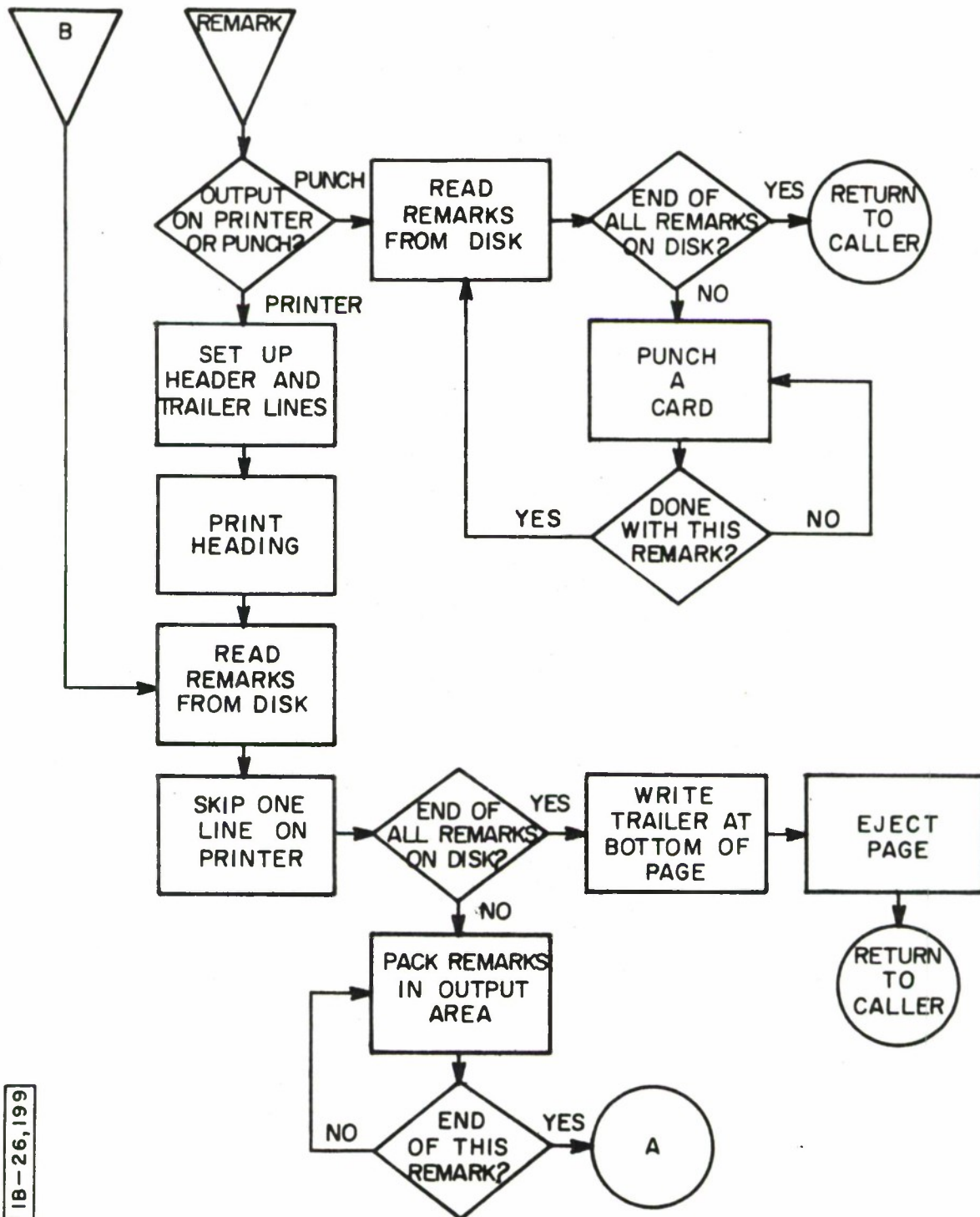
Description

During the input or file generation phase of REPGEN, any input records defined as remarks are put into a remarks area on the system disk. Remarks normally would be punched into cards, and a given remark may not exceed five cards in length.

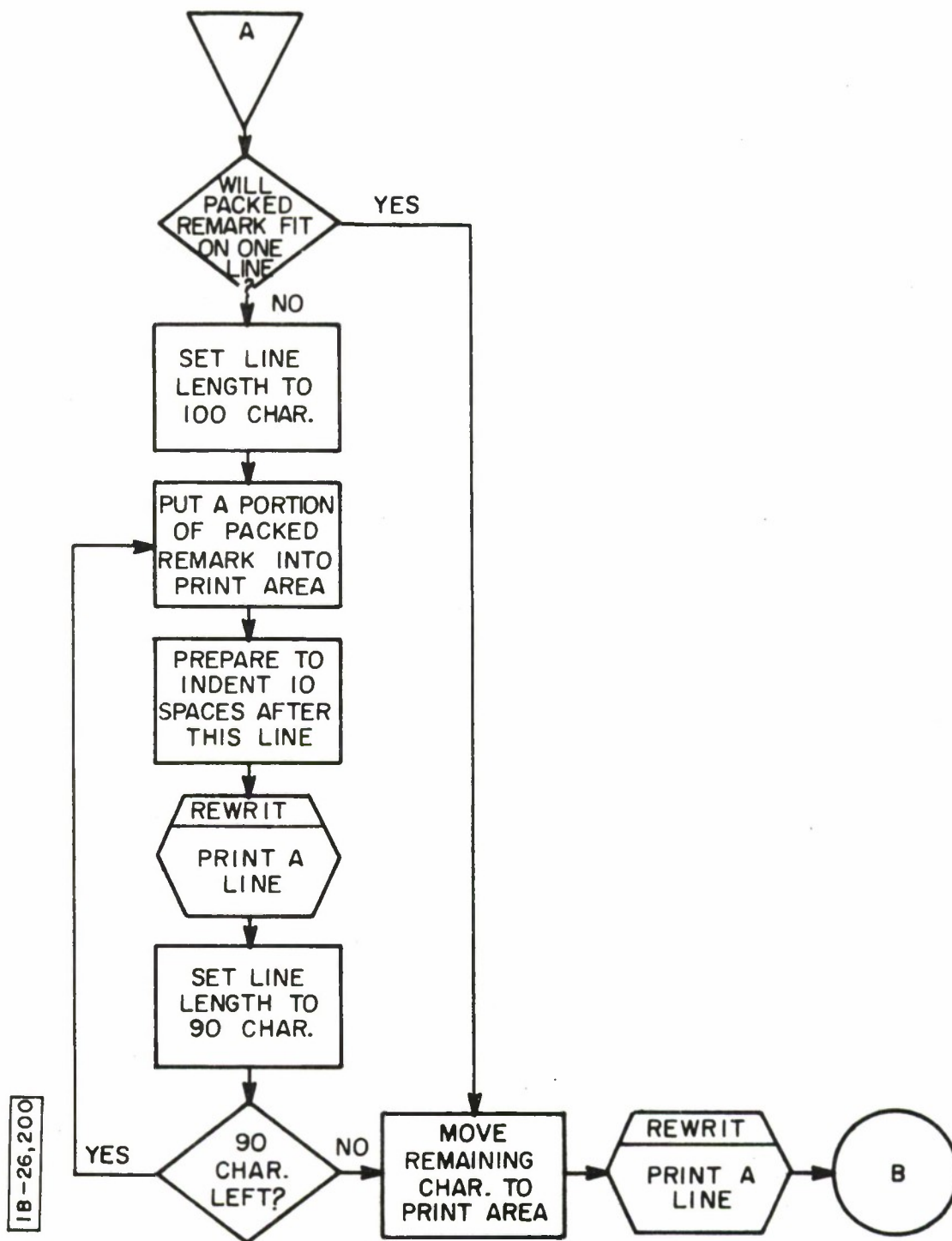
If the output medium is punched cards, the REMARK processor simply reads back the remarks stored on disk and punches them into cards.

If the output medium is the printer, REMARK prints a heading and reads a remark from the disk. The remark is packed into a print line and printed. If it overflows to more than one line, subsequent lines are indented ten spaces until the entire remark is printed. The printer is spaced once, the next remark is read and the process is repeated until all remarks have been printed.

The REMARK routine is called and executed after REALO has finished executing an output program.



IB-26,199



REPEET

Purpose

A stored query GETTAC statement is retrieved, expanded and executed.

Calling Sequence

REPEET is called internally by REPGEN. The call is placed in SDA 0003 as

REPEETNNN

where NNN is the saved statement number assigned by SAVER. REPGEN then issues a

B COPS
DCW 'REPEET'

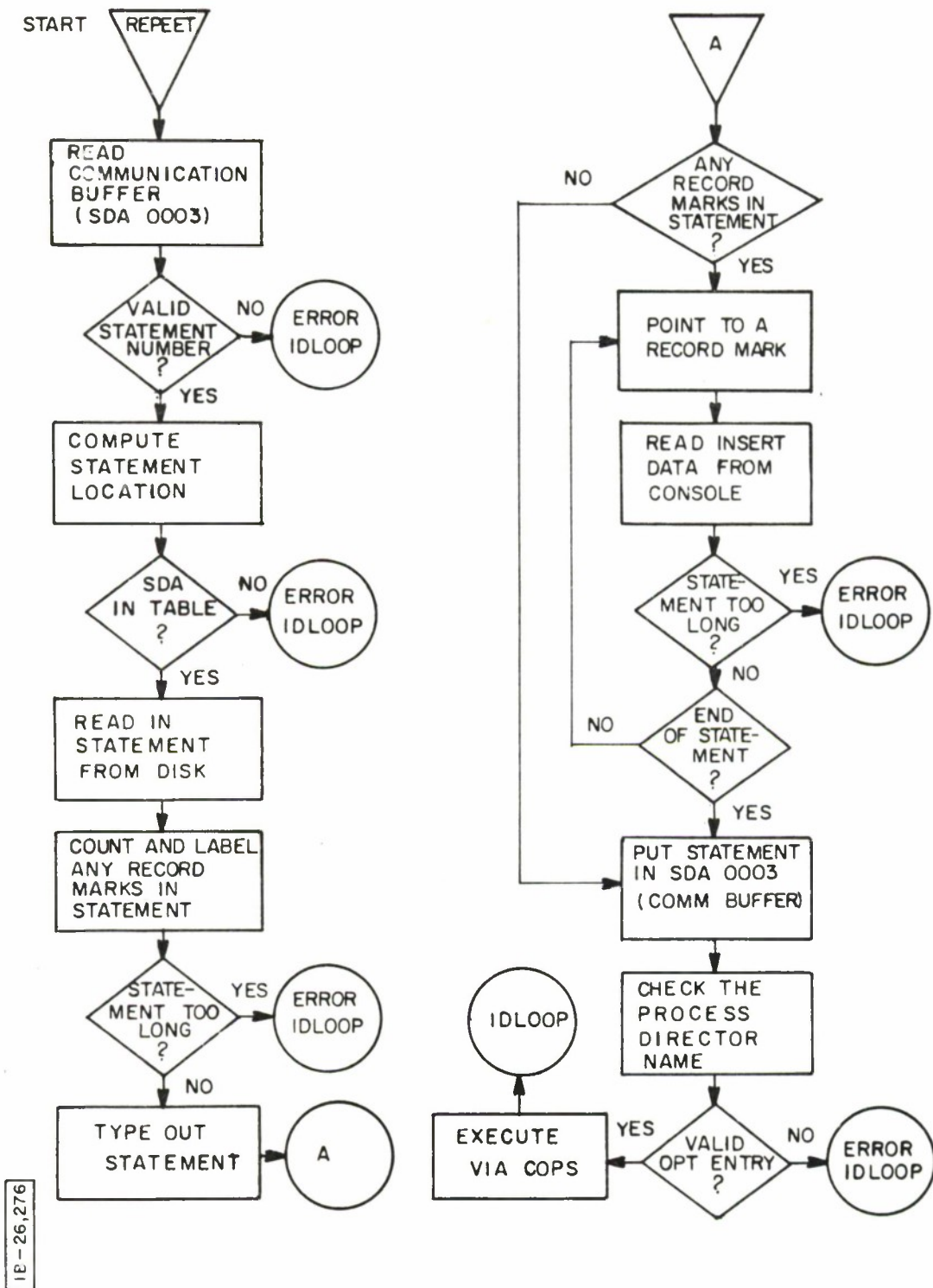
REPEET may be called externally by typing

REPEETNNN

on the console typewriter.

Description

The statement number is validity checked. The saved statement SDA table is read in from disk and the SDA corresponding to the given statement number is extracted. The saved statement is read into core from disk. The statement is scanned for record marks. Each record mark is labeled by a record mark number. The statement is then typed out and data insertions are made via typewriter for each record mark in the statement. The edited statement is written to the system communication area and the GETTAC processor is called via COPS.



REKALL

Purpose

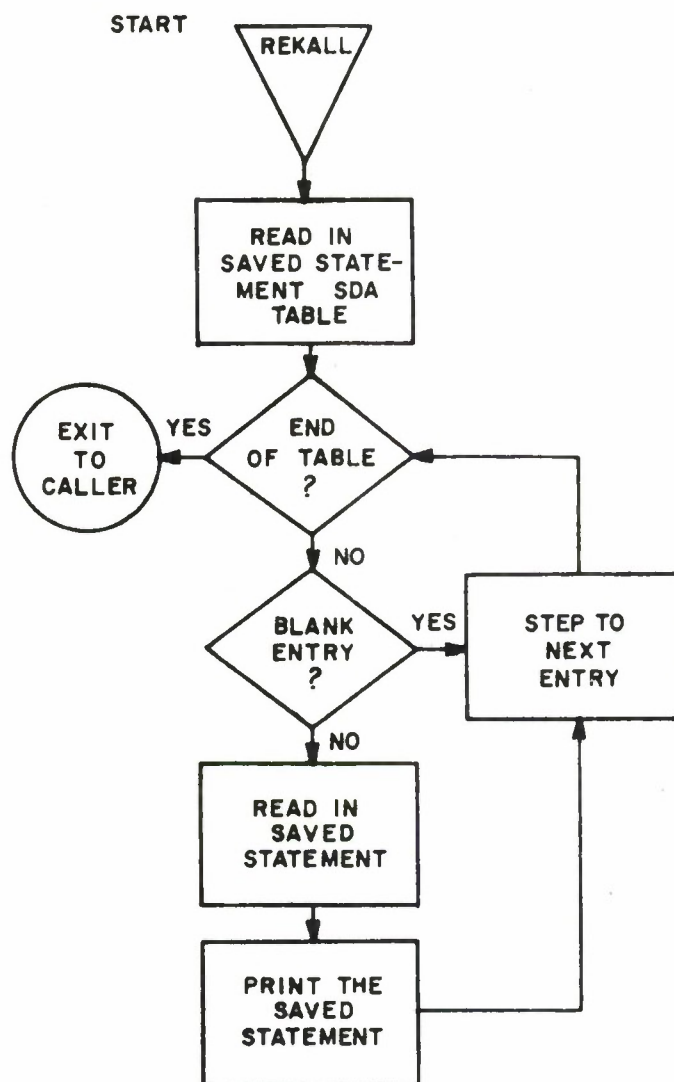
Print out all statments previously stored on disk by the program SAVER.

Calling Sequence

Enter REKALL on the console typewriter. Return will be to IDLOOP.

Description

REKALL reads the saved statement SDA table into core. Each non-blank entry points to a saved statement. Each saved statement is read in turn from the disk and is printed along with its assigned statement number. A total of 500 statements may be saved.



1A-26,233

DELEET

Purpose

Delete from the system disk a Query Language statement previously stored there by the SAVER program.

Calling Sequence

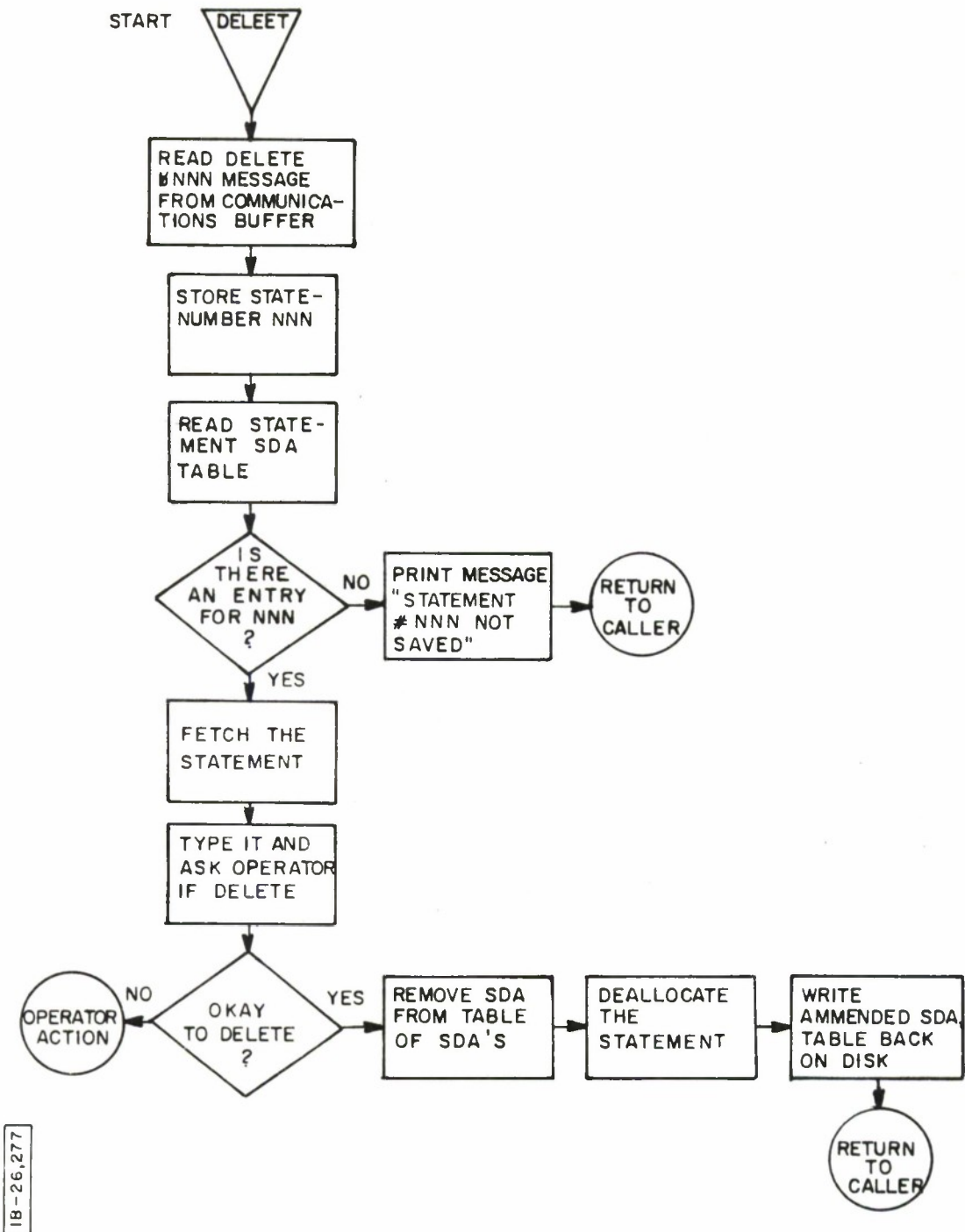
Enter on the console typewriter

DELEETNNN

where NNN is the saved statement number assigned by SAVER.

Description

The DELEETNNN message is retrieved from SDA 0003. The 3 character number referencing the statement to be deleted is stored. The table of 4 character SDA's which references where each saved statement is stored is read into core. The first entry in the table references statement number 001, the second entry references statement number 002, and so on. Blank entries indicate that statement number has not been saved and is available. The statement SDA to be deleted is then located. If it is blank, a message will be printed on the console indicating that this statement cannot be deleted because it has not been saved. If an SDA is found in the table entry corresponding to the statement number to be deleted, the saved statement is typed out on the I/O typewriter for the operator's approval. If okay, the SDA is deleted from the statement table and also deallocated so that the statement itself is effectively deleted. The new SDA statement table is now rewritten on the disk and the program returns to IDLOOP.



PRTCRD

Purpose

PRTCRD is used to print a report from cards which were generated by RPTCRD.

Calling Sequence

Enter PRTCRD on the console typewriter. Return will be to IDLOOP.

Description

The print area is cleared, the printer is spaced to a new page, and a card is read.

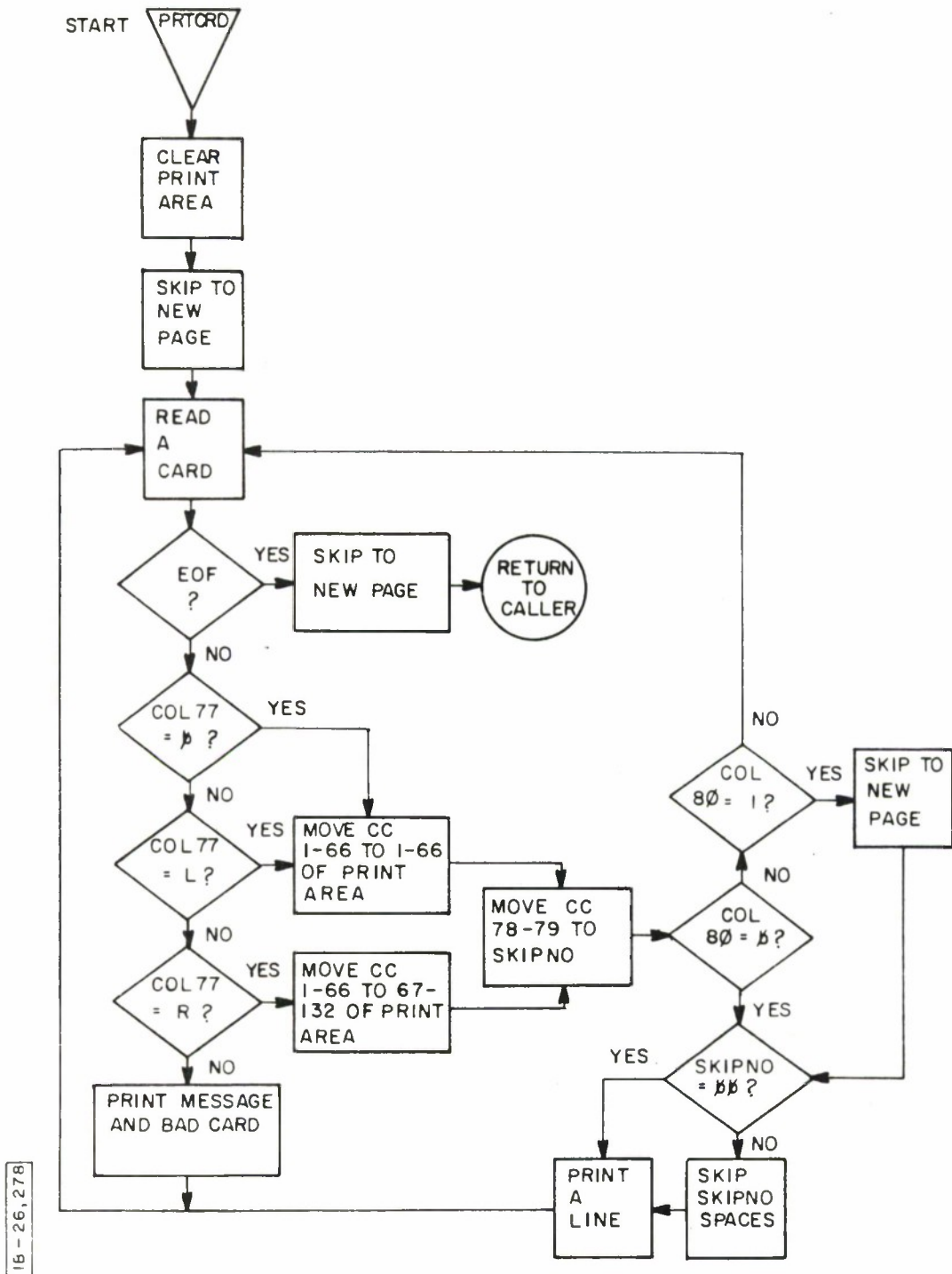
Each input card contains half of a print image in columns 1-66. Columns 77-80 contain information identifying which half of the print image the card represents and any page or line spacing to be done before printing. Columns 77-80 are interpreted as follows:

Column 77	= L or R	; This is left half of print image.
		; This is right half of print image.

Columns 78-79	= nn	; Number of times to space before printing.
		Blank specifies single spacing.

Column 80	= %	; Print after putting this card in buffer and performing required spacing.
	1	; Same as %, except skip to new page first.
	*	; Put this card in buffer but do not print or space.

As each card is read, columns 1-66 are placed into the left or right half of a print image. If column 80 is a blank or 1, required spacing is performed and a line is printed. If column 80 is *, no printing or spacing is done, and the next card is immediately read.



PRTTAP

Purpose

PRTTAP is used to print additional copies of a report from a REPGEN extra-copies tape.

Calling Sequence

PRTTAP is called internally from RGPOUT. RGPOUT puts

PRTTAPn

where n is the number of copies desired, into SDA 0003 and then calls

```
B      COPS
DCW    'PRTTAP'
```

The program can also be called externally by entering

PRTTAPn

on the console typewriter. Return will be to IDLOOP.

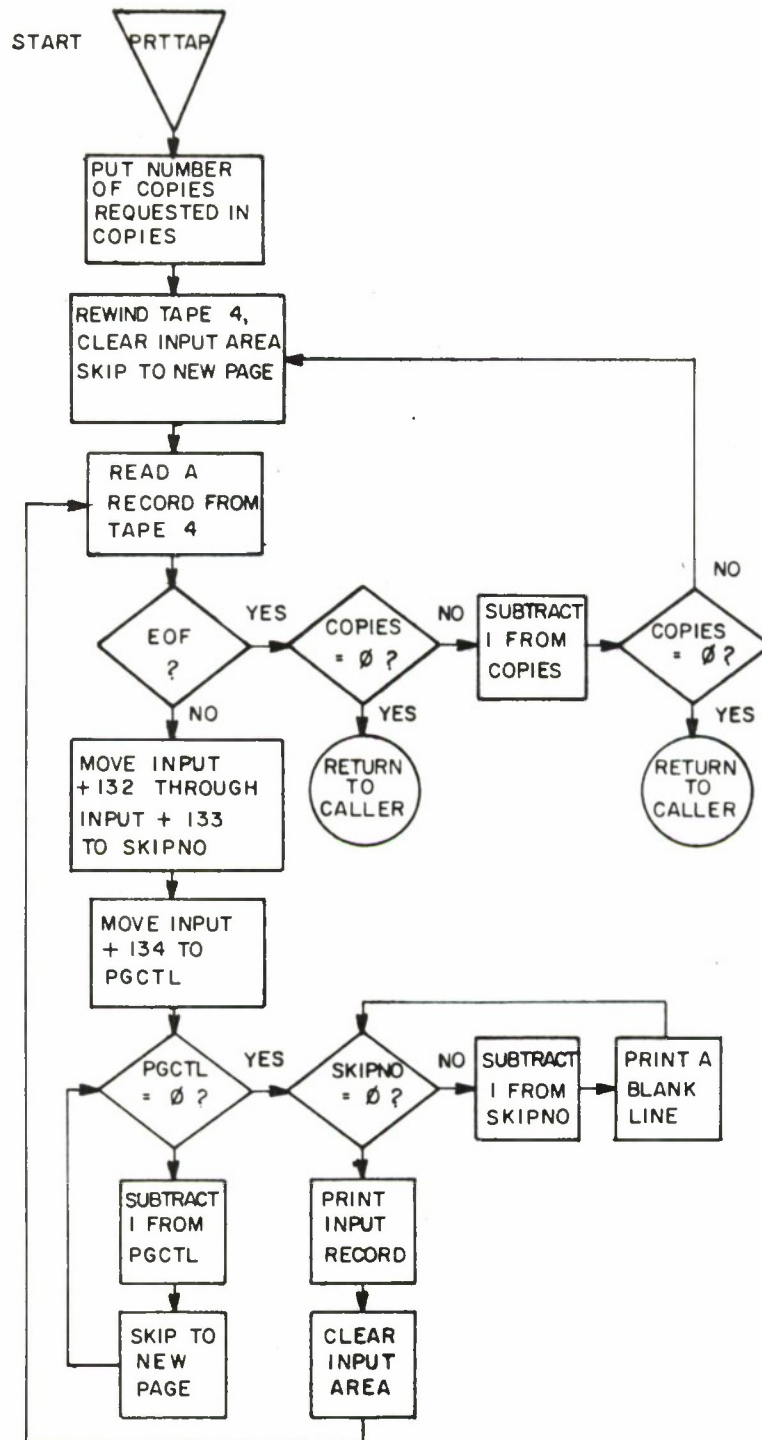
Description

The input to PRTTAP is tape 4, with one file of 135 character records written in even parity. The first 132 characters of each record represent a print image. The last three characters of each record contain information regarding page and line spacing to be accomplished before printing the image.

The format of each record is as follows:

Character	1-132	= Print image.
	133-134	= Number of times to space before printing.
		Blank specifies single spacing
	135	= Page control.
		1 means skip to new page before printing; blank means do not skip.

PRTTAP reads a record and first examines the page control character. If it is a 1, a skip to channel 1 is executed. Next, if any spacing is indicated it is accomplished by printing the appropriate number of blank lines. The line is then printed. When an end-of-file is encountered, the tape is rewound, and the number of copies is decremented by one. If it is not blank, another run is made through the tape.



IB-26,279

RPTCRD

Purpose

RPTCRD is used to punch cards from a REPGEN extra-copies tape in a format which will allow the cards to be processed by PRTCRD.

Calling Sequence

Enter RPTCRD on the console typewriter. Return will be to IDLOOP.

Description

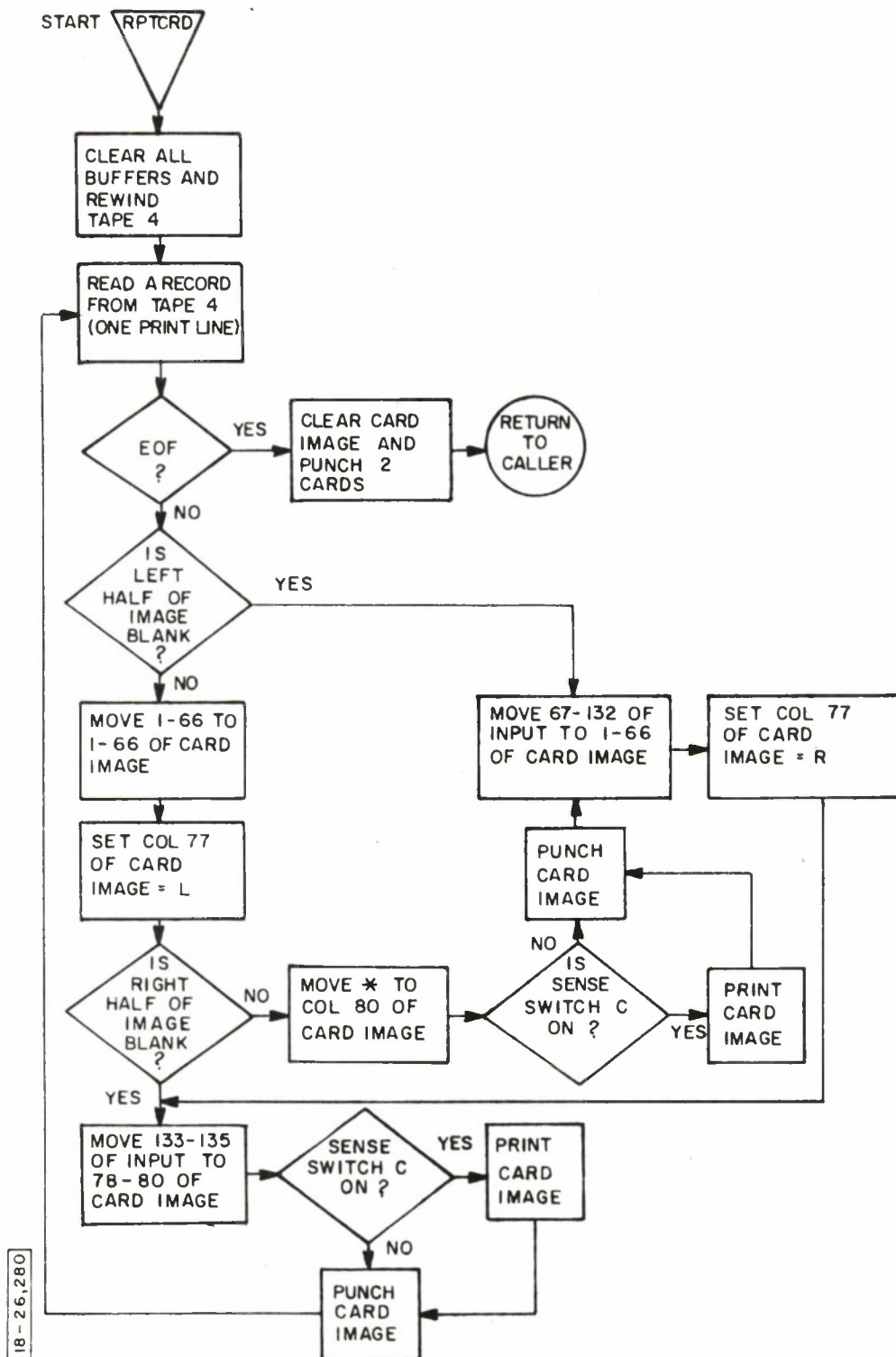
Input to RPTCRD is tape 4, with one file of 135 character records written in even parity.

Each input record from tape 4 is treated as a print line, where the first 132 characters constitute a print image, and the last three characters are for carriage control. (See write-up of PRTTAP for discussion of the carriage control characters.)

RPTCRD will divide each print image into a left and right half, and punch each half into the first 66 columns of a card using columns 77-80 of the cards for carriage control information. These cards are in proper format to be processed by PRTCRD, which will use these cards to generate a report on the printer. If Sense Switch C is on, each card will be printed as well as punched.

The format of the output cards and the meaning of the data entered in columns 77-80 is the same as that which was presented in section 6.18.3 to describe the input cards to PRTCRD.

When an end-of-file is encountered on tape 4, two blank cards are punched and control returns to IDLOOP.



DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) The MITRE Corporation Bedford, Massachusetts		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	
		2b. GROUP N/A	
3. REPORT TITLE PROGRAM/CODING SPECIFICATION FOR THE AFICCS REPORT GENERATION (REPGEN) CAPABILITY			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) N/A			
5. AUTHOR(S) (First name, middle initial, last name) Norman B. Sutherland			
6. REPORT DATE January 1969		7a. TOTAL NO. OF PAGES 170	7b. NO. OF REFS 0
8a. CONTRACT OR GRANT NO. AF 19(628)-5165		9a. ORIGINATOR'S REPORT NUMBER(S) ESD-TR-68-134	
b. PROJECT NO. 512V		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) MTR-564	
c.			
d.			
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES N/A		12. SPONSORING MILITARY ACTIVITY Director of Planning and Technology, Electronic Systems Division, Air Force Systems Command, L. G. Hanscom Field, Bedford, Mass.	
13. ABSTRACT The Report Generation (REPGEN) capability comprises a set of programs designed to run on an IBM 1410 computer under the Air Force Integrated Command and Control System (AFICCS). The primary purpose of REPGEN is to provide a capability for producing a printed report from a file of input data contained on punched cards, magnetic tape or on the system disk. This document contains detailed information regarding the structure and operation of the REPGEN programs, including table descriptions, core maps, program flow-charts and abstracts.			

KEY WORDS

LINK A

LINK B

LINK C

ROLE

WT

ROLE

WT

ROLE

WT

REPORT GENERATOR

AFICCS REPORT GENERATION CAPABILITY

REPORT GENERATOR PROGRAM SPECI-
FICATIONS - (1410)